Bundesamt für Sicherheit in der Informationstechnik

Zertifikat

BSI-ITS-0003-1991

mit

Zertifizierungsbericht

zu

SINIX-S V5.22

der

Siemens-Nixdorf Informationssysteme AG

BSI - Bundesamt für Sicherheit in der Informationstechnik, Godesberger Allee 183
5300 Bonn 2, Postfach 200363, Tel. 0228-9582-0, FAX 0228-9582-400

Vorbemerkung zur Rechtsgrundlage

Das Bundesamt für Sicherheit in der Informationstechnik (BSI) hat gemäß §3 und §4 BSIG[1)] die Aufgabe, Produkte der Informationstechnik im Hinblick auf ihre Sicherheitseigenschaften zu prüfen und zu bewerten.

Das BSI schließt eine solche Prüfung (Evaluierung) mit einem Sicherheitszertifikat (zusammenfassende Bewertung) und einem Zertifizierungsbericht ab. Zertifikat und Bericht bilden eine Einheit. Der Bericht enthält

- die sicherheitstechnische Beschreibung des evaluierten Produktes,
- die Einzelheiten der Bewertung (Stärken und Schwächen),
- Auflagen an den Betrieb.

1) Gesetz über die Errichtung des Bundesamtes für die Sicherheit in der Informationstechnik (BSIG) vom 17. Dezember 1990, BGBl. I S.2834

Zertifizierung

1. Prüfgrundlage

Das Produkt SINIX-S V5.22[2)] der Siemens Nixdorf Informationssysteme AG wurde auf der Basis

- der IT-Sicherheitskriterien in der 1. Fassung vom 11. Januar 1989[3)],
- des IT-Evaluationshandbuchs in der 1. Fassung vom 22. Februar 1990[4)]

evaluiert.

2. Durchführung der Prüfung

Die Evaluierung wurde durchgeführt von Rainer Bertelsmaier, Stollmann[5)], Cornelia Heinicke, Stollmann, Heinrich Kersten, BSI[6)], Heribert Spindler, IABG[7)], Markus Ullmann, BSI, Marcel Weinand, BSI, Ridvan Ysiltepe, Stollmann.
Die vom Evaluationsteam vorgelegten Prüfberichte entsprechen in Form und Inhalt den Anforderungen der genannten Prüfgrundlagen.

---

2) SINIX™ ist das UNIX-Produkt der Siemens Nixdorf Informationssysteme AG
UNIX™ ist ein eingetragenes Warenzeichen der UNIX System Laboratories, Inc.

3) IT-Sicherheitskriterien: Kriterien für die Bewertung der Sicherheit von Systemen der Informationstechnik(IT), ISBN 3-88784-192-1, Bundesanzeiger-Verlag Köln(1989), und GMBl. 1989, S. 277

4) IT-Evaluationshandbuch: Handbuch für die Prüfung der Sicherheit von Systemen der Informationstechnik(IT), ISBN 3-88784-220-0, Bundesanzeiger-Verlag Köln(1990), und GMBl. 1990, S. 430

5) Stollmann GmbH

6) Bundesamt für Sicherheit in der Informationstechnik

7) Industrieanlagen-Betriebsgesellschaft mbH

3. Gültigkeit des Zertifikats

Das Zertifikat gilt nur für die angegebene Version des Produktes. Jede Änderung an sicherheitsrelevanten Teilen fällt nicht mehr unter das Zertifikat.
Die Gültigkeit des Zertifikats kann aber auf neue Versionen und Releases des Produktes ausgedehnt werden, sofern der Hersteller eine Nachevaluierung des geänderten Systems entsprechend den Vorgaben des IT-Evaluationshandbuchs (Abschnitt 9) beantragt und die Prüfung keine sicherheitstechnischen Mängel ergibt.
Das Zertifikat gilt nur unter der Voraussetzung, daß

- alle Auflagen bzgl. Generierung, Konfiguration und Betrieb, soweit sie im nachfolgenden Bericht angegeben sind, beachtet werden,
- das Produkt auf der angegebenen Hardware und - sofern angegeben - in der beschriebenen Umgebung betrieben wird.

4. Veröffentlichung

Der nachfolgende Zertifizierungsbericht enthält die Seiten 1-66. Das Produkt SINIX-S V5.22 ist in die Liste der zertifizierten Produkte aufgenommen worden. Exemplare dieses Berichtes können bei allen Geschäftsstellen des Herstellers[8)] des Produktes angefordert werden.

Bonn, 26.07.1991

(Dr. Mertz)

8) Siemens Nixdorf Informationssysteme AG, Otto-Hahn-Ring 6, 8000 München 83

Bundesamt für Sicherheit in der Informationstechnik

Zertifikat

Das Betriebssystem SINIX-S V5.22 der Siemens Nixdorf Informationssysteme AG wurde nach den IT-Sicherheitskriterien (GMBl. 1989, S. 277) und dem IT-Evaluationshandbuch (GMBl. 1990, S. 430) evaluiert.
Das evaluierte Produkt ist ein Betriebssystem, das auf Rechnern des Typs MX300 (MX300, MX300-05, MX300-10, MX300-15 und MX300-30) - jeweils Standard- und Tempest-Version - mit den Prozessoren NS32332 und NS32532 lauffähig ist.
Es wird für die Erstinstallation auf zwei Disketten (Bezeichnungen SINIX-H V522: SINIX0 V522 und SINIX1 V522) und drei Tape-Cartridges (SINIX V522N10, SINIXK V522A00 N2, SINIX-S V5.22 N10.5) ausgeliefert. Zum Lieferumfang gehört weiterhin eine aus mehreren Bänden bestehende Systemdokumentation (siehe Zertifizierungsbericht Gliederungspunkt 1.2).

Die höchste Funktionalitätsklasse, in der das Produkt **alle** Anforderungen erfüllt, ist die

Klasse F1.

Darüber hinaus sind für die Grundfunktionen Identifikation und Authentisierung, Rechteprüfung und Beweissicherung die Anforderungen der **Klasse F2** erfüllt.

Das Produkt SINIX-S V5.22 erreichte die

Qualitätsstufe Q2.

Dieses Zertifikat gilt nur in Verbindung mit dem zugehörigen Zertifizierungsbericht. Die in diesem Bericht aufgeführten Auflagen an den Einsatz des Produktes sind zu beachten.

Bonn, 26.07.1991

(Dr. Mertz)

Zertifizierungsbericht

SINIX-S V5.22

Gliederung des Zertifizierungsberichtes

1. Beschreibung des evaluierten Systems 3
   - 1.1 System-Konfiguration 3
   - 1.2 Liste der zum evaluierten System gehörenden Anwenderdokumentation 12
2. Globale Beschreibung der Sicherheitsanforderungen des evaluierten Systems 14
   - 2.1 Identifikation und Authentisierung 15
   - 2.2 Rechteverwaltung 17
   - 2.3 Rechteprüfung 18
   - 2.4 Beweissicherung 19
   - 2.5 Wiederaufbereitung 19
3. Beschreibung der Evaluation mit Hinweisen auf kritische Bereiche 21
   - 3.1 Qualität der Sicherheitsanforderungen 21
   - 3.2 Qualität der Spezifikation 21
   - 3.3 Qualität der verwendeten Mechanismen 21
   - 3.4 Qualität der Abgrenzung zu nicht zu evaluierenden Systemteilen 54
   - 3.5 Qualität des Herstellungsvorgangs 57
   - 3.6 Betriebsqualität 59
   - 3.7 Qualität der Anwenderdokumentation 61
4. Weitere Hinweise und Auflagen 62

1. Beschreibung des evaluierten Systems

Bei dem geprüften System handelt es sich um das Betriebssystem SINIX-S V5.22. SINIX-S unterscheidet sich vom Basisprodukt SINIX durch die Portierung des Paketes SMP (*Security Modul Package* der Fa. SecureWare) und durch weitere herstellerspezifische Verbesserungen zur Erhöhung der Sicherheit.
Das evaluierte Produkt ist ein Betriebssystem, das auf Rechnern des Typs MX300 (MX300, MX300-05, MX300-10, MX300-15 und MX300-30) - jeweils Standard- und Tempest-Version - mit den Prozessoren NS32332 und NS32532 lauffähig ist.
Als Benutzerschnittstellen unterstützt das System die Schnittstellen X/OPEN (XPG III) und UNIX BSD4.2; die Systemschnittstelle bedient außerdem Systemaufrufe des Systems SINIX V2.1.
Das System kann in zwei unterschiedlichen Betriebsarten gefahren werden:

- Ein - Benutzer - Modus: Der Zugriff ist nur für den *Super User* (Systemverwalter) erlaubt.
- Mehr - Benutzer - Modus: Eine Anmeldung ist von allen angeschlossenen Terminals aus möglich.

1.1 System-Konfiguration

Das evaluierte Betriebssystem SINIX-S V5.22 sieht eine feste Software-Konfiguration vor. Es ist im Rahmen des evaluierten Systems **nicht möglich** oder vorgesehen, nach dem Baukastenprinzip zusätzliche Software-Teile zum Kern dazuzubinden.
Der Kern */vmunix* enthält bereits alle Treiber für alle von SNI angebotenen zusätzlichen Geräte (*SCSI-Harddisk, Ethernet Controller* usw.). Beim Hinzufügen von Geräten, d.h. Änderung der Hardware-Konfiguration, werden die zugehörigen Treiber beim Booten des Systems automatisch aktiviert. Aufgrund dieses Sachverhaltes ist mit der Festlegung der evaluierten Software-Konfiguration auch die Hardware-Konfiguration bestimmt.
Die Zertifizierung des Betriebssystems SINIX-S V5.22 gilt nur für die Hardware-Konfiguration, bei der ausschließlich Treiber aktiviert sind, die zum zertifizierten System gehören.

Zum zertifizierten System gehören die Treiber:

mm	*Memory Device Driver*
sy	*/dev/tty TTY Driver*
sw	*Swap Device Driver*
lo	*Loopback Interface Driver*
sl	*Serial Line Internet Protocol*
vt	*Collage TTY Driver*
ptc, pts	*Pseudo Terminal Device Driver*
aud	*Audit Device Driver*
tic	*Ticket Device Driver*
is, it	*Storager Device Driver*
sr	*SERAG Device Driver*
mc	*Real Time Clock*

Die erlaubten Konfigurationsmöglichkeiten beschränken sich auf den Anschluß zusätzlicher Terminals oder eines Druckers an die serielle Schnittstelle.

Nicht zum zertifizierten System gehören die Treiber:

vp	*Virtual Partitions*
sc	*SIGNETICS SC2681 DUART*
ex	*Exos 201 Ethernet Controller*
ly	*Lync Graphic Board*
si, sv	*SIM Board*
nc	*NCR SCSI Controller*
hd	*SCSI Disk*
lad	*Juke Box*
exa	*Exabyte*
trx	*TRACEX*
psr	*Testdriver*
bam	*BS2000 Interface*
cmx,cd,cx	*CMX Driver* (kein */dev*-Eintrag)
cl, cw	*CMX Driver* (kein */dev*-Eintrag)
seq	*CMX Driver* (kein */dev*-Eintrag)

s_0 *DUEAI Controller*
xt *XYLOGICS 472 Tape Controller*
*due_, sl_* *DUEAI/DUXAI Board* (kein */dev*-Eintrag)

Prüfsummen für sicherheitsrelevante Systemteile:

Bei Installation des Systems SINIX-S V5.22 werden zu jeder eingespielten Datei die Dateigröße, die Zeit des letzten Zugriffs und der letzten Änderung, sowie eine Prüfsumme ausgegeben und in die Datei */C2install.out* protokolliert.
Der Anwender kann durch den Vergleich der protokollierten Prüfsummen mit den hier aufgelisteten Prüfsummen sicherstellen, daß er das evaluierte Produkt installiert hat.

Requested: `/.bin/df`
Filesize: 24576
Last acc.: Thu Mar 7 15:38:35 1991
Mod. tim.: Thu Apr 26 01:26:59 1990
Checksum for file `/.bin/df` is: 2194340

Requested: `/.bin/lpr`
Filesize: 118784
Last acc.: Thu Mar 7 15:16:58 1991
Mod. tim.: Wed May 2 08:35:50 1990
Checksum for file `/.bin/lpr` is: 10131950

Requested: `/.bin/mai1`
Filesize: 65536
Last acc.: Thu Mar 7 14:13:19 1991
Mod. tim.: Thu Apr 26 01:27:21 1990
Checksum for fi1e `/.bin/mai1` is: 5482376

Requested: `/.bin/passwd`
Filesize: 147456
Last acc.: Thu Mar 7 15:59:01 1991
Mod. tim.: Mon Mar 4 14:25:45 1991
Checksum for fi1e `/.bin/passwd` is: 12261249

Requested: */.bin/ps*
Filesize: 131072
Last acc.: Thu Mar 7 15:59:00 1991
Mod. tim.: Mon Mar 4 14:25:38 1991
Checksum for file */.bin/ps* is: 11255491

Requested: */.bin/su*
Filesize: 126976
Last acc.: Thu Mar 7 15:59:02 1991
Mod. tim.: Mon Mar 4 14:25:49 1991
Checksum for file */.bin/su* is: 10870344

Requested: */.bin/write*
Filesize: 102400
Last acc.: Thu Mar 7 15:59:03 1991
Mod. tim.: Mon Mar 4 14:25:58 1991
Checksum for file */.bin/write* is: 8772070

Requested: */etc/atcronif*
Filesize: 196608
Last acc.: Thu Mar 7 15:56:32 1991
Mod. tim.: Mon Mar 4 14:46:46 1991
Checksum for file */etc/atcronif* is: 16710279

Requested: */etc/auth/dlvr_audit*
Filesize: 77824
Last acc.: Thu Mar 7 15:56:30 1991
Mod. tim.: Mon Mar 4 14:51:14 1991
Checksum for file */etc/auth/d1vr_audit* is: 6579386

Requested: */etc/sysadmif*
Filesize: 204800
Last acc.: Thu Mar 7 15:56:33 1991
Mod. tim.: Mon Mar 4 14:46:49 1991
Checksum for fi1e */etc/sysadmif* is: 17323057

Requested: */tcb/bin/audinit*
Filesize: 61440
Last acc.: Thu Mar 7 15:56:54 1991
Mod. tim.: Mon Mar 4 14:51:06 1991
Checksum for file */tcb/bin/audinit* is: 5148617

Requested: */tcb/bin/auditcmd*
Filesize: 69632
Last acc.: Thu Mar 7 15:56:52 1991
Mod. tim.: Mon Mar 4 14:50:54 1991
Checksum for file */tcb/bin/auditcmd* is: 5702037

Requested: */tcb/bin/auditd*
Filesize: 77824
Last acc.: Thu Mar 7 15:56:56 1991
Mod. tim.: Mon Mar 4 14:50:56 1991
Checksum for file */tcb/bin/auditd* is: 6735232

Requested: */tcb/bin/auditif*
Filesize: 249856
Last acc.: Thu Mar 7 15:56:50 1991
Mod. tim.: Mon Mar 4 14:46:37 1991
Checksum for file */tcb/bin/auditif* is: 21178472

Requested: */tcb/bin/authck*
Filesize: 131072
Last acc.: Thu Mar 7 15:56:53 1991
Mod. tim.: Mon Mar 4 14:50:59 1991
Checksum for file */tcb/bin/authck* is: 11170117

Requested: */tcb/bin/authif*
Filesize: 249856
Last acc.: Thu Mar 7 15:56:36 1991
Mod. tim.: Mon Mar 4 14:46:33 1991
Checksum for file */tcb/bin/authif* is: 21227658

Requested: */tcb/bin/checkfile*
Filesize: 45056
Last acc.: Thu Mar 7 16:05:42 1991
Mod. tim.: Mon Mar 4 14:51:00 1991
Checksum for file */tcb/bin/checkfile* is: 3888240

Requested: */tcb/bin/chg_audit*
Filesize: 28672
Last acc.: Thu Mar 7 15:56:52 1991
Mod. tim.: Mon Mar 4 14:46:43 1991
Checksum for file */tcb/bin/chg_audit* is: 2227378

Requested: */tcb/bin/init*
Filesize: 61440
Last acc.: Thu Mar 7 15:56:35 1991
Mod. tim.: Mon Mar 4 14:27:23 1991
Checksum for file */tcb/bin/init* is: 5412065

Requested: */tcb/bin/integrity*
Filesize: 94208
Last acc.: Thu Mar 7 15:56:54 1991
Mod. tim.: Mon Mar 4 14:51:04 1991
Checksum for fi1e */tcb/bin/integrity* is: 8053302

Requested: */tcb/bin/reduce*
Filesize: 106496
Last acc.: Thu Mar 7 15:56:55 1991
Mod. tim.: Mon Mar 4 14:51:09 1991
Checksum for file */tcb/bin/reduce* is: 9020261

Requested: */tcb/bin/report*
Filesize: 131072
Last acc.: Thu Mar 7 15:56:51 1991
Mod. tim.: Mon Mar 4 14:46:41 1991
Checksum for file */tcb/bin/report* is: 11044517

Requested: */tcb/bin/run_reduce*
Filesize: 32768
Last acc.: Thu Mar 7 15:56:51 1991
Mod. tim.: Mon Mar 4 14:46:38 1991
Checksum for file */tcb/bin/run_reduce* is: 2977943

Requested: */tcb/bin/startaudit*
Filesize: 283
Last acc.: Thu Mar 7 15:56:56 1991
Mod. tim.: Mon Mar 4 12:15:35 1991
Checksum for file */tcb/bin/startaudit* is: 23935

Requested: */tcb/bin/stopaudit*
Filesize: 133
Last acc.: Thu Mar 7 15:56:56 1991
Mod. tim.: Mon Mar 4 12:15:35 1991
Checksum for file */tcb/bin/stopaudit* is: 11806

Requested: */tcb/lib/getty*
Filesize: 40960
Last acc.: Thu Mar 7 15:58:16 1991
Mod. tim.: Mon Mar 4 14:27:20 1991
Checksum for file */tcb/lib/getty* is: 3072702

Requested: */tcb/lib/initcond*
Filesize: 106496
Last acc.: Thu Mar 7 15:58:17 1991
Mod. tim.: Mon Mar 4 14:52:00 1991
Checksum for file */tcb/1ib/initcond* is: 8886175

Requested: */tcb/lib/login*
Filesize: 147456
Last acc.: Thu Mar 7 15:58:15 1991
Mod. tim.: Mon Mar 4 14:26:03 1991
Checksum for file */tcb/lib/login* is: 12623918

Requested: */usr/.bin/mesg*
Filesize: 118784
Last acc.: Thu Mar 7 15:58:45 1991
Mod. tim.: Mon Mar 4 14:36:46 1991
Checksum for file */usr/.bin/mesg* is: 10371747

Requested: */usr/att/bin/ipcs*
Filesize: 126976
Last acc.: Thu Mar 7 15:58:42 1991
Mod. tim.: Mon Mar 4 14:36:37 1991
Checksum for file */usr/att/bin/ipcs* is: 10836743

Requested: */usr/att/bin/mail*
Filesize: 114688
Last acc.: Thu Mar 7 14:36:11 1991
Mod. tim.: Wed Apr 25 22:35:40 1990
Checksum for file */usr/att/bin/mail* is: 9745248

Requested: */usr/att/bin/mesg*
Filesize: 118784
Last acc.: Thu Mar 7 16:06:41 1991
Mod. tim.: Mon Mar 4 14:36:46 1991
Checksum for file */usr/att/bin/mesg* is: 10371747

Requested: */usr/att/bin/mkdir*
Filesize: 45056
Last acc.: Thu Mar 7 15:16:21 1991
Mod. tim.: Wed Apr 25 22:35:43 1990
Checksum for file */usr/att/bin/mkdir* is: 3659557

Requested: */usr/att/bin/newgrp*
Filesize: 122880
Last acc.: Thu Mar 7 15:58:44 1991
Mod. tim.: Mon Mar 4 14:36:44 1991
Checksum for file */usr/att/bin/newgrp* is: 10461707

Requested: */usr/att/bin/rmai1*
Filesize: 114688
Last acc.: Thu Mar 7 16:06:46 1991
Mod. tim.: Wed Apr 25 22:35:40 1990
Checksum for file */usr/att/bin/rmail* is: 9745248

Requested: */usr/att/bin/rmdir*
Filesize: 45056
Last acc.: Thu Mar 7 15:16:19 1991
Mod. tim.: Wed Apr 25 22:35:44 1990
Checksum for file */usr/att/bin/rmdir* is: 3632809

Requested: */usr/att/usr/bin/at*
Filesize: 151552
Last acc.: Thu Mar 7 15:58:38 1991
Mod. tim.: Mon Mar 4 14:36:27 1991
Checksum for file */usr/att/usr/bin/at* is: 12828324

Requested: */usr/att/usr/bin/crontab*
Filesize: 126976
Last acc.: Thu Mar 7 15:58:41 1991
Mod. tim.: Mon Mar 4 14:36:33 1991
Checksum for file */usr/att/usr/bin/crontab* is: 10737174

Requested: */usr/att/usr/1ib/mai1x/rmmai1*
Filesize: 36868
Last acc.: Thu Mar 7 14:37:41 1991
Mod. tim.: Wed Apr 25 21:53:59 1990
Checksum for file */usr/att/usr/1ib/mai1x/rmmai1* is: 2842201

Requested: */usr/etc/quot*
Filesize: 57344
Last acc.: Thu Mar 7 14:32:21 1991
Mod. tim.: Thu Apr 26 01:08:27 1990
Checksum for file */usr/etc/quot* is: 5149294

Requested: */vmunix*
Filesize: 1218860
Last acc.: Thu Mar 7 16:05:40 1991
Mod. tim.: Thu Mar 7 16:05:33 1991
Checksum for file */vmunix* is: 103745209

1.2 Liste der zum evaluierten System gehörenden Anwender-dokumentation

Anwenderbezogene Dokumentation des Basis-Systems:

Zur Installation und zum Betrieb von SINIX V5.22 sind Manuale erforderlich. Die Manuale sind Bestandteil der sprachabhängigen Dokumentationspakete:

SINIX-DOC-D V5.22	Dokumentation für deutsche Version
SINIX-DOC-GB V5.22	Dokumentation für englische Version

Zum Lieferumfang der Dokumentationspakete gehören:

Dokumentenbezeichnung:	Bestellnummer:
Betriebsanl. MX300 und CAD-Arbeitsplatz 9733	U3905-J-Z95-3
SINIX Systemverwaltung V5.22	U3901-J-Z95-4
SINIX V5.22 Kommandos Teil1	U5453-J1-Z95-2
SINIX V5.22 Kommandos Teil2	U5454-J1-Z95-2
SINIX V5.22 Kommandos Teil3	U5455-J1-Z95-2
C-Entwicklungssystem V5.22 Teil1	U3899-J-Z95-3
C-Entwicklungssystem V5.22 Teil2	U3900-J-Z95-3
SINIX-Spool Benutzerhandbuch	U5650-J-Z95-1
COLLAGE-Bediensystem V5.22 (inkl. Kurzbeschr.)	U5995-J-Z95-1
Bediensystem Kurzbeschreibung	U6007-J-Z95-1
COLLAGE Bedienen Verwalten Programmieren	U3004-J-Z95-4
Nachtrag COLLAGE Bedienen Verwalten Programmieren	U3004-J1-Z95-5
SINIX Buch1	U3201-J-Z95-l
SINIX Buch2	U3202-J-Z95-1

Nachtrag SINIX Buch2	U3202-J1-Z95-2
SINIX Systemsicherheit	U5069-J-Z95-1
NLS Internationalization in SINlX (englisch)	U5098-J-Z95-2-7600

Mit SINIX V5.22 ist die Bedienoberfläche für den *Super User* auf COLLAGE umgestellt. Damit sind die entsprechenden Kapitel (z.B. Systemadministration) im SINIX Buch2 ungültig.

Anwenderbezogene Dokumentation (SINIX-S V5.22-spezifisch)

Dokumentenbezeichnung:	Bestellnummer:
Sicherheitspaket Benutzerhandbuch	U5971-J-Z95-1
Sicherheitspaket Systemverwaltung	U5973-J-Z95-1
MX300 (SINIX-S) Betriebsanleitung Nachtrag	U3905-J-Z95-4
MX300 (SINIX-S) Systemverwaltung Nachtrag	U3904-J1-Z95-5

2. Globale Beschreibung der Sicherheitsanforderungen des evaluierten Systems

Bestandteil der Sicherheitsanforderungen sind die Grundfunktionen:

- Identifikation und Authentisierung
- Rechteverwaltung
- Rechteprüfung
- Beweissicherung
- Wiederaufbereitung

Der diese Grundfunktionen zur Verfügung stellende sicherheitsrelevante und zu evaluierende Teil des Systems besteht aus

- dem Systemkern (mit Ausnahme solcher Treiber, die nicht zur evaluierten Konfiguration gehörende Komponenten unterstützen),
- den sicherheitsrelevanten Programmen (SUID/SGID-Programmen) und den
- sicherheitsrelevanten Dateien.

Die sicherheitsrelevanten Programme und Dateien werden in separaten Dateibäumen gehalten und sind durch die Mechanismen der Rechteverwaltung / Rechteprüfung gegen den Zugriff von unprivilegierten Benutzern geschützt. Die Verwaltung der für die Sicherheitsfunktionen relevanten Dateien erfolgt im Rahmen von Subsystemen, für die besondere Benutzer-Privilegien erforderlich sind, vermittelt durch menügesteuerte Bediener-Schnittstellen.
Folgende Subsysteme stehen zur Verfügung:

- *Memory Subsystem*: Es schützt die Datenstrukturen der Benutzerprozesse und des Systemkerns vor unberechtigtem Zugriff.
- *Terminal Subsystem*: Es reguliert die Lese- und Schreibzugriffe auf die einzelnen angeschlossenen Terminals.
- *Authentication Subsystem*: Es stellt eine Reihe von Diensten für die Verwaltung und Auswertung von Authentisierungsdaten zur Verfügung.
- *Audit Subsystem*: Hierin sind alle Funktionen zur Beweissicherung zusammengefaßt.
- *Cron Subsystem*: Es ist für die Abwicklung von Batch-Aufträgen zuständig.

- *Sysadmin Subsystem*: Hierin werden Verwaltungsdienste für den Betrieb des Rechners zur Verfügung gestellt.

Als Subjekte im System werden Benutzer bzw. Rollen und Prozesse betrachtet. Mehrere Benutzer können zu Gruppen mit eigenen Rechten zusammengefaßt werden. Ein Benutzer kann Mitglied in maximal acht Gruppen sein.

Zur Administration sicherheitsrelevanter Funktionen sieht das System mehrere Administrator-Rollen vor, mit denen Zugangsprivilegien zu den zugehörigen Subsystemen verbunden sind.

- *Authentication Administrator*: Verwaltung der Authentisierungsdaten
- *Audit Administrator*: Verwaltung der Protokollierung und Auswertung sicherheitsrelevanter Ereignisse
- *Cron Administrator*: Verwaltung der Berechtigung der Batch-Aufträge
- *Super User* (Systemverwalter): Benutzer mit uneingeschränkten Rechten; verfügt als einziger Benutzer über drei verschiedene Kennungen (*root*, *admin*, *shutdown*) mit gleicher Kurzkennung 0

Objekte sind Dateiobjekte (normale Dateien, Datei-Verzeichnisse, Gerätedateien, Named Pipes), die IPC-Objekte "Message Queue" und "Semaphore" und in bestimmten Situationen auch Prozesse.

Aktionen sind Zugriffe von Subjekten auf der Rechteverwaltung unterliegende Objekte. Die möglichen Aktionen werden in den nächsten Abschnitten beschrieben.

Die Kommunikationsendpunkte "Sockets" unterliegen als Objekte im System **nicht** der Rechteverwaltung.

2.1 Identifikation und Authentisierung

Der Zugang zum System ist nur solchen Benutzern möglich, für die eine Benutzerkennung (Name) und eine Kurzkennung (RUID: Reale Benutzer-Kurzkennung, Zahlenwert zwischen 0 und 9999), sowie entsprechende Gruppenkennungen

eingetragen sind. Bei Verwendung des Menüsystems *authif* zur Eintragung von Benutzern ist eine Eindeutigkeit der Eintragungen sichergestellt.

Das System SINIX-S V5.22 identifiziert und authentisiert alle Benutzer einschl. des *Super Users*, bevor irgendeine andere Interaktion des Systems mit dem Benutzer stattfindet. Auch für ein erfolgreiches Hochfahren in den Ein-Benutzer-Modus muß sich der *Super User* an der Konsole identifizieren und authentisieren.

Der Zugang zum System erfordert die Identifikation durch Angabe der Benutzerkennung und Authentisierung durch die (nicht sichtbare) Eingabe des Paßwortes, das im System verschlüsselt und an einer für den normalen Benutzer nicht zugänglichen Stelle abgespeichert ist. Das System bietet verschiedene Möglichkeiten für die Paßwortvergabe (Wahl des Paßwortes, Überprüfung des Paßwortes auf Palindrom, Erzwingung eines Paßwortwechsels). Mehrfache erfolglose Authentisierungsversuche führen zum Sperren der Benutzerkennung.
Bei erfolgreicher Authentisierung wird für den Benutzer ein Prozeß mit einem für ihn hinterlegten Programm (i.a. ein Kommando-Interpreter "Shell") gestartet, der mit der Identität und dadurch mit den Rechten des Benutzers ausgestattet ist. Alle folgenden Aktivitäten des Benutzers im System werden durch diesen Prozeß und seine Sohnprozesse realisiert. Der Prozeß bekommt eine unveränderliche Kennung (LUID: Login Benutzer-Kurzkennung) zugewiesen, die an alle von ihm erzeugten Prozesse weitervererbt wird. Anhand der LUID sind alle Aktivitäten des Benutzers eindeutig identifizierbar.

Die für die Rechteprüfung relevanten Identitäten eines Prozesses können sich unter bestimmten Bedingungen in der Prozeßkette ändern, z.B. bei Ausführung sog. SUID / SGID-Programme.

Die sensitiven Paßwortdaten und die für die Benutzerrechte und -privilegien relevanten Benutzerdaten stehen in Dateien, die gegen Zugriff unprivilegierter Benutzer geschützt sind. Auf diese Dateien haben nur der *Super User* und der *Authentication Administrator* Zugriff.

Nach einem Logout besteht für den betreffenden Benutzer keine I/O - Verbindung mehr zum System.

2.2 Rechteverwaltung

Die "effektiven" und die "realen" Identitäten werden für den Benutzerprozeß beim Login mit der entsprechenden Benutzer- und Gruppen-Kurzkennung belegt. Im Laufe des Betriebs können sich die "effektiven" Identitäten (EUID und EGID) von den "realen" Identitäten (RUID und RGID) unterscheiden.

Die Zugriffsrechte an Dateiobjekten und den IPC-Objekten "Message Queue" und "Semaphore" sind gegeben

- durch die Kurzkennungen des Benutzers bzw. seiner Gruppe und den daraus resultierenden "effektiven" Identitäten der von ihm gestarteten Prozesse einerseits,
- sowie durch die im *Permission Mode Word* (PMW) des Objekts vermerkten für den Zugriff erforderlichen Rechte andererseits.

Die Rechtestruktur am Objekt unterscheidet zwischen Lese-, Schreib- und Ausführungsrechten und dort jeweils zwischen Eigentümer, Gruppe und anderen Benutzern. Beim "Einhängen" eines Dateisystems durch *mount*, das durch den *Super User* erfolgen kann, ist es möglich, die Zugriffsart zu Dateien global auf den Lesezugriff zu beschränken.

Bei IPC-Objekten gibt es zusätzlich zum Eigentümer einen Erzeuger und eine entsprechende Gruppe; ein dem Ausführungsrecht entsprechendes Recht ist für IPC-Objekte nicht definiert.

Es wird generell zwischen expliziten und impliziten Rechten unterschieden. Erstere werden vom Benutzer den Objekten zugewiesen und stehen in Relation zur Identität des Benutzers. Implizite Rechte sind Rechte, die in der Prozeßkette eines Benutzers vermittelt über einen Wechsel der Identität ausgeübt werden können (i.a. bei Ausführung eines SUID / SGID-Programms).

Außer den Zugriffsrechten können im PMW von Dateiobjekten die sog. "s-Bits" (*setuid, setgid*) für Eigentümer oder Gruppe sowie das "t-Bit" (*sticky bit*) vergeben werden. Das "s-Bit" besagt, daß die (Programm-)Datei als SUID- bzw. SGID-Programm ausgeführt wird, d.h. mit den Rechten (der "effektiven" Identität) des **Dateieigentümers** bzw. der Benutzergruppe des Objektes. Das *sticky bit* bewirkt

bei einem Datei-Verzeichnis, daß Dateien darin nur durch den Dateieigentümer oder den *Super User* gelöscht werden können.

Nur der Eigentümer und der *Super User* dürfen die Zugriffsrechte an Objekten ändern. Auch die Gruppe und der Eigentümer eines Objekts können nur durch Eigentümer oder *Super User* geändert werden; der Eigentümer braucht zusätzlich ein spezielles Systemprivileg (*chown*). Für IPC-Objekte gelten die Rechte gleichermaßen für Eigentümer und Erzeuger. Der Erzeuger bzw. die Erzeugergruppe eines IPC-Objekts sind unveränderlich.
"s-Bits" können nur vom *Super User* oder dem Dateieigentümer vergeben werden; der Eigentümer braucht zusätzlich ein spezielles Systemprivileg (*chmodsugid*).
Das "t-Bit" für ein Verzeichnis wird nur vom *Super User* vergeben, kann aber vom Eigentümer des Verzeichnisses zurückgesetzt werden.

Neben dieser Standard-Rechteverwaltung sieht das System eine Reihe von Privilegien vor, die für spezielle Aktionen zusätzlich gegeben sein müssen. Es wird unterschieden zwischen sog. Systemprivilegien, die in einer zur Prozeßstruktur parallelen Struktur verwaltet werden, und Kommandoprivilegien. Systemprivilegien berechtigen einen Prozeß zu speziellen Aktivitäten im Systemkern. Kommandoprivilegien sind zur Ausführung einiger sicherheitsrelevanter Kommandos oder Kommandoteile erforderlich.
Die Informationen für die Zuweisung der Privilegien an Benutzer sind Teil der geschützten Authentisierungsdaten. Die Privilegien können nur durch den *Super User* oder den *Authentication Administrator* vergeben werden.
Ein Benutzer kann nicht mehr Kommando- und Systemprivilegien erlangen, als ihm durch die Systemadministration zugewiesen wurden, auch nicht durch Ausführung von SUID / SGID-Programmen.

2.3 Rechteprüfung

Die Prüfung der Rechte, die im PMW der Objekte gehalten werden, erfolgt in den Systemaufrufen, die die Öffnung von Objekten realisieren. Dabei muß die intendierte Zugriffsart angegeben werden. Bei Zugriffen auf geöffnete Dateien werden die Benutzerrechte nicht mehr geprüft. Beim Lese- oder Schreibzugriff wird vielmehr nur überprüft, ob die Datei für diese Zugriffsart geöffnet wurde.

Ist das erforderliche Recht für einen Zugriff nicht gegeben, so wird der Systemaufruf abgebrochen. Der Rückgabewert des Aufrufs besagt, daß ein Fehler vorliegt; der Fehlercode wird in einer Fehlervariablen übergeben.

Systemprivilegien werden in den entsprechenden Systemaufrufen geprüft.

Kommandoprivilegien werden in den zugehörigen Kommandos geprüft. Die dafür erforderlichen Informationen werden aus der geschützten Datenbasis, in der die Benutzer mit dem jeweiligen Kommandoprivileg verzeichnet sind, gelesen. Die Konsequenzen einer negativen Prüfung hängen von der Art des Kommandos ab: Abbruch oder eingeschränkte Funktionalität des Kommandos.

2.4 Beweissicherung

Das System verfügt über eine aufwendige Protokoll-Komponente. Der *Audit Administrator* besitzt vielfältige Möglichkeiten, die Erzeugung von Protokolldaten bzgl. Art, Auswahl und Menge der zu protokollierenden Daten zu steuern. Zur Auswertung der Protokolldaten bietet das System eine Reihe von Werkzeugen.
Die Zuordnung der Aktionen zu den Benutzern ist eindeutig durch die beim Systemzugang vergebene, nicht änderbare LUID möglich.
Der normale Benutzer kann weder die Erzeugung von Audit-Daten beeinflussen, noch Informationen stattgefundener Ereignisse fälschen. Darüber hinaus hat er keinen Zugriff zu den Programmen und Daten der Auswertung.
Durch den Mechanismus der Beweissicherung wird auch der *Super User* kontrolliert.

2.5 Wiederaufbereitung

Das Löschen von Dateien auf dem Plattenspeicher wird grundsätzlich über die logische Freigabe der betroffenen Datenblöcke realisiert; der Inhalt selbst wird dabei nicht physikalisch gelöscht. Ein spezielles Kommando ermöglicht aber, den Dateiinhalt zu überschreiben.

Der Zugriff auf Dateien erfolgt über Routinen des Systemkerns und über dessen Puffer; ein normaler Benutzer (ohne Zugriffsrecht auf das Speichermedium) kann

nur auf den Inhalt existierender Dateien bzw. auf von ihm selbst erzeugte Daten zugreifen. Ein Zugriff auf die Inhalte schon gelöschter Dateien ist nur möglich, wenn die Datei während des Löschens noch geöffnet war und nur solange, bis die Datei geschlossen wird.

Die Dateieinträge in Verzeichnissen werden beim Löschen der Datei logisch entfernt, bleiben aber physikalisch erhalten.

Die Seiten des Arbeitsspeichers werden bei der Zuweisung an den Benutzer gelöscht (binär Null).

Der direkte Zugriff auf Datenträger und Speicher ist durch die Mechanismen der Rechteverwaltung geschützt.

Der Bildschirm der Datensichtstation wird nach dem Beenden der Sitzung mit einem einheitlichen "Begrüßungsbildschirm" für die Anmeldung des nachfolgenden Benutzers überschrieben.

3. Beschreibung der Evaluation mit Hinweisen auf kritische Bereiche

3.1 Qualität der Sicherheitsanforderungen

Das vorgelegte Dokument "Betriebssystem SINIX MX300 - Sicherheitsanforderungen" ist in natürlicher Sprache verfaßt. Es beschreibt für die für Funktionalitätsklasse F2 geforderten Grundfunktionen jeweils die Sicherheitsanforderungen und die zugehörigen Mechanismen und stellt den Bezug zu den Bedrohungen her.
Die Sicherheitsanforderungen entsprechen mit Ausnahme der Grundfunktionen "Wiederaufbereitung" und "Rechteverwaltung" den in den IT-Sicherheitskriterien für die Funktionalitätsklasse F2 formulierten Anforderungen. Die fehlende Wiederaufbereitung der Datenträger (Diskette, Magnetband-Kassette) und die Granularität der Rechteverwaltung entsprechen nicht den Anforderungen der Funktionalitätsklasse F2.

3.2 Qualität der Spezifikation

Trotz einiger Mängel (kein einheitlicher struktureller Rahmen, keine hierarchische Ableitung und Zuordnung) ist das vorliegende Dokument noch geeignet, um die Sicherheitsanforderungen für die angestrebte Qualitätsstufe überprüfen zu können.
Die Prüfung des Systems bzgl. der Unumgehbarkeit der Sicherheitsfunktionen wurde nur an den Stellen durchgeführt, an denen aufgrund der Spezifikation Schwachstellen vermutet wurden. Dabei wurden keine Mängel festgestellt.
Die Spezifikation entspricht den Anforderungen der Stufe Q2.

3.3 Qualität der verwendeten Mechanismen

Gegenstand der Mechanismenprüfung sind in erster Linie prinzipielle Schwächen von Mechanismen, die diesen unabhängig von der Implementierung inhärent sind.
Die Gesamtbewertung der Mechanismen ergibt sich aus den Einzelbewertungen der unter 3.3 aufgeführten Mechanismen. Die Stufe **mittelstark** wurde durchgän

gig erreicht. Damit entsprechen die Mechanismen in ihrer Gesamtheit den Anforderungen der Qualitätsstufe Q2.

3.3.1 Qualität der Mechanismen zur Identifikation und Authentisierung

Mechanismus zur Identifikation und Authentisierung von Benutzern:

Die Identifikation und Authentisierung von Benutzern erfolgt über die Angabe einer Benutzerkennung und des zugehörigen Paßwortes (Authentisierung durch Wissen). Das Paßwort wird einwegverschlüsselt und mit dem in einer Datei des Systems hinterlegten verschlüsselten Paßwort verglichen. Fällt dieser Vergleich positiv aus, wird ein mit den Rechten des Benutzers ausgestatteter Prozeß - als für den Benutzer im System agierende Einheit - erzeugt.

Durch den implementierten Paßwort-Alterungsmechanismus kann der *Authentication Administrator* über Wahl des minimalen Paßwort-Änderungszeitraums (*minimum change time*), der Paßwort-Verfallzeit (*expiration time*) und der Paßwort-Lebensdauer (*life time*) den zeitlichen Gültigkeitsbereich von Paßwörtern einschränken. Der normal berechtigte Benutzer kann diesen Mechanismus nicht umgehen.

Um Probierattacken zu begegnen, kann der *Authentication Administrator* sowohl für Terminals als auch für Kennungen eine Obergrenze für die Zahl der erlaubten aufeinanderfolgenden erfolglosen Login-Versuche unter dieser Kennung bzw. an diesem Terminal angeben, nach deren Erreichen die Kennung bzw. das Terminal gesperrt wird. Diese Sperre kann anschließend nur vom *Authentication Administrator* oder dem *Super User* aufgehoben werden.
Wenn die Zahl der Login-Fehlversuche begrenzt ist, ist ein normaler Benutzer in der Lage, eine Sperrung von Benutzern, deren Kennung ihm bekannt ist, herbeizuführen.

Bei der Änderung bzw. erstmaligen Eingabe eines Paßworts wird das angegebene Paßwort nach verschiedenen Kriterien auf seine Güte hin geprüft:

- Falls bei der Eintragung des Benutzers durch den *Authentication Administrator* die Einstellung "*password restricted*" gewählt wurde, wird das Paßwort daraufhin geprüft, ob es vom Kennungs- oder Gruppennamen abgeleitet ist, ein Palindrom ist oder in einer Liste mit "verbotenen"

Paßwörtern vorkommt (z.B. Firmenbezeichnungen, Vornamen, Monatsnamen o.ä.).
- Es erfolgt eine Prüfung darauf, ob das neue Paßwort ungleich dem alten Paßwort ist.

Zur Absicherung der Kommunikationsverbindung Benutzer / System beim Login wird bei der Paßwort-Eingabe (beim Login oder bei der Paßwortänderung) das Echo unterdrückt. Darüber hinaus werden nach dem Logout I/O-Verbindungen eventueller Hintergrundprozesse des Benutzers zum Terminal gekappt, um Versuche eines Hintergrundprozesses, Daten vom Terminal zu lesen, zu begegnen. Weiterhin werden aus einer geschützten Datenbasis beim neuerlichen Login die Zeitpunkte des letzten erfolgreichen und erfolglosen Logins ausgegeben, wodurch der Benutzer in Erfahrung bringen kann, ob ihm eventuell von einem Spoofing-Programm ein erfolgloser Login-Versuch vorgetäuscht wurde, um das Benutzer-Paßwort auszuspähen.

Nach erfolgreicher Prüfung der Zulässigkeit des Logins und der Korrektheit des Paßworts wird ein Prozeß erzeugt, der innerhalb des Systems für den Benutzer agiert. Die Verwaltungsstruktur dieses Prozesses wird dabei mit Parametern aus der *Authentication Database* für diesen Benutzer besetzt, insbesondere mit den effektiven und realen Benutzer- und Gruppen-Kurzkennungen (EUID: Effektive Benutzer-Kurzkennung, RUID: Reale Benutzer-Kurzkennung, EGID: Effektive Gruppen-Kurzkennung, RGID: Reale Gruppen-Kurzkennung) sowie der Login Benutzer-Kurzkennung (LUID).
Damit ist weitestgehend sichergestellt, daß sich
- die im System agierenden Prozesse und
- Aktionen, die durch die Prozesse auf Veranlassung der Benutzer ausgeführt werden,

eindeutig einem Benutzer im Sinne einer realen Person zuordnen lassen.

Der *Authentication Administrator* landet nach erfolgreichem Login im *Authentication Administrator Interface* (*authif*). Dadurch kann er in der *Authentication Database* nur in einer vom Programm kontrollierten Art und Weise Modifikationen vornehmen.

Mechanismus zur Identifikation von Prozessen:

Prozesse werden vom System mittels verschiedener Kennungen identifiziert:
- reale und effektive Benutzer-Kurzkennung (RUID, EUID)
- reale und effektive Gruppen-Kurzkennung (RGID, EGID)
- Login Benutzer-Kurzkennung (LUID)
- Prozeßidentifikation (PID)
- Prozeßidentifikation des Vaterprozesses (PPID)
- Prozeßgruppenidentifikation

Die Kennungen RUID, EUID, RGID, EGID und die LUID werden dem Login-Prozeß zugeordnet und bei Erzeugen weiterer Prozesse durch diesen Prozeß vererbt (Ausnahme SUID- / SGID-Programme).
Die anderen Kennungen werden bei der Erzeugung eines Prozesses gesetzt. Dabei wird die PID vom System eindeutig vergeben, die PPID der PID des erzeugenden Prozesses und die Prozeßgruppen-ID der PID des Prozeßgruppenvaters gleichgesetzt.
Die Kennungen LUID und PID sind für den normalen Benutzer unveränderbar, können aber vom *Super User* durch schreibenden Zugriff auf die Prozeß-Datenstrukturen[9)] im Hauptspeicher verändert werden (evtl. auch in manipulativer Absicht durch ein "Trojanisches Pferd").

3.3.2 Qualität der Mechanismen zur Rechteverwaltung

Mechanismus "*Discretionary Access Control* (DAC)*"* für Dateiobjekte:

Dateien sind dem System durch ihren Namen (Pfad), der bei der Erzeugung der Datei vergeben wird, bekannt. Durch die Bildung von Verweisen ist es möglich, daß eine Datei über mehrere Namen angesprochen werden kann. Der Name wird innerhalb des Kerns mittels Durchlaufen der im Dateipfad angegebenen Verzeichnisse auf eine eindeutige *inode*-Nummer abgebildet. Der Zugriff auf die Datei erfolgt dann über diesen *inode*.
In Programmen werden geöffnete Dateien durch eine Kurzkennung (Deskriptor) identifiziert. Der Deskriptor weist auf einen Eintrag in der Tabelle der offenen

9) Der Eingriff in die Prozeß-Datenstrukturen ist nur mit Aufwand möglich und erfordert eingehende Systemkenntnisse.

Dateien des Benutzers. Der Eintrag in der Tabelle zeigt auf den zugehörigen *inode*.

Die Zugriffsrechte an Dateiobjekten werden im *Permission Mode Word* (PMW) des Objekts im Dateisystem gehalten. Zugriffsrechte können getrennt vergeben werden für den Eigentümer des Objekts, die Gruppe des Objekts und alle anderen Benutzer. Es ist nicht möglich, die Zugriffsrechte an einem Objekt für jeden Benutzer getrennt zu vergeben. Aufgrund dieser Eigenschaft ist der Mechanismus nicht geeignet, Funktionalität gemäß der Funktionalitätsklasse F2 zu realisieren.
Es wird unterschieden zwischen Lese-, Schreib- und Ausführ- bzw. Suchrecht. Der Zugriff kann also auf einen nichtmodifizierenden Zugriff beschränkt werden.
Die Zugriffsrechte eines Subjekts an einem Dateiobjekt sind gegeben durch

- die Identität des Subjektes (das sind die effektive Benutzer-Kurzkennung und die effektive Gruppen-Kurzkennung des zugreifenden Prozesses) und
- durch die Identität des Eigentümer bzw. der Gruppe und das PMW des Objekts.

Die Zugriffsrechte im PMW werden beim Anlegen des Objekts durch den Eigentümer gesetzt; dies erfolgt über Systemaufrufe, die von System-Kommandos (Kommandos, die vom Hersteller mit dem System ausgeliefert werden) oder Anwenderprogrammen aufgerufen werden. Die initialen Zugriffsrechte eines Benutzers sind in seiner Benutzerumgebung (*environment*) gesetzt und können dort von ihm oder vom *Super User* geändert werden. Beim Objekt werden als Eigentümer und Eigentümer-Gruppe die effektiven Kurzkennungen des erzeugenden Prozesses eingetragen.
Die Änderung der Zugriffsrechte im PMW kann sowohl durch den Eigentümer als auch den *Super User* erfolgen; sie geschieht ebenfalls mittels Systemaufrufen im Rahmen von System-Kommandos oder Anwenderprogrammen.
Durch Ändern des Eigentümers eines Dateiobjektes können die Rechte an dem Objekt indirekt verändert werden. Dies ist allerdings nur dem Eigentümer oder dem *Super User* möglich. Der Eigentümer benötigt zusätzlich das Systemprivileg *chown*.
Bei der Änderung von Rechten gilt für die Prüfung jeweils das zuletzt vergebene Recht.

Für den Zugriff auf eine Datei sind nicht nur Rechte an der Datei selbst, sondern auch das Suchrecht für die Verzeichnisse des Dateipfads erforderlich. Durch *link* besteht die Möglichkeit, Dateien über verschiedene Zugriffspfade zugänglich zu machen, wobei unterschiedliche Zugriffsrechte möglich sein können. Diese Umstände konstituieren aber keinen Widerspruch, da bei jedem Zugriff der Zugangspfad vom Benutzer bzw. seiner Benutzerumgebung (*environment*) spezifiziert wird, und da in jedem Fall eindeutig geregelt ist, ob der Benutzer über diesen Pfad auf die Datei zugreifen kann. Es kann der Fall eintreten, daß der Eigentümer einer Datei auf diese nicht mehr zugreifen kann.
Der *link*-Mechanismus kann zu unübersichtlichen Rechteverhältnissen führen. Der Eigentümer einer Datei hat i.d.R. keine Möglichkeit festzustellen, wo die Links seiner Dateien liegen und wer sie erzeugt hat.

Die Gruppenrechte im PMW geben nur eine eingeschränkte Information; sie sagen zwar etwas darüber aus, welche Zugriffsrechte eine Gruppe von Benutzern hat, aber nicht, welche Benutzer dieser Gruppe angehören oder Zugang zu ihr haben. Die Gruppenzugehörigkeit der Benutzer kann außerdem wechseln. Die Dateien, die alle Informationen über die Gruppenzugehörigkeiten enthalten, sind von jedem Benutzer lesbar.

Ein Subjekt, das in einem Verzeichnis mit Schreib- und Suchberechtigung ausgestattet ist, kann dort beliebig Dateien, auch solche, deren Besitzer er nicht ist, löschen und neue Dateien mit identischem Namen wieder einrichten. Besitzt dieses Subjekt dazu noch das Systemprivileg *chown*, so ist es ihm möglich, die neu eingerichteten Dateien mit den ursprünglichen Eigentümern zu versehen. Auf diese Weise ist das Einbringen von "Trojanischen Pferden" möglich. Um diese Möglichkeit zu unterbinden, darf anderen Subjekten das Schreibrecht auf anderen Subjekten gehörende Verzeichnisse nicht erteilt werden.

Bei einem Umbenennen oder Verschieben von Dateiobjekten bleiben die alten Zugriffsrechte am Dateiobjekt erhalten. Durch Veränderung des Dateipfads können sich aber die Zugriffsmöglichkeiten ändern.

Beim Ausscheiden eines Benutzers bleiben seine Zugriffsrechte erhalten. Das Menüsystem zur Benutzerverwaltung löscht den Benutzer nicht aus der Benutzer

datei, sondern sperrt ihn lediglich, so daß Benutzerkennung und Kurzkennung im System erhalten bleiben. Beim Eintragen neuer Benutzer sucht das System eine unbenutzte Benutzerkennung, so daß vermieden wird, daß eine alte oder doppelte Benutzerkennung, zu der schon / noch Zugriffsrechte an Dateiobjekten vorhanden sind, vergeben wird. Dieser Schutz ist bei Umgehung des Menüsystems nicht gegeben.

Die Ausübung der an Dateien eines Dateisystems gegebenen Zugriffsrechte setzt voraus, daß das Dateisystem durch den Systemaufruf *mount*, zu dem nur der *Super User* berechtigt ist, den Benutzern zugänglich gemacht wurde. Dabei besteht die Möglichkeit, den Zugang auf Lesezugriffe zu beschränken, wodurch die existierenden Schreibrechte an den Dateien des Dateisystems für die Dauer des Anschlusses (bis zum *umount* und erneutem *mount*) außer Kraft gesetzt sind. Diese Möglichkeit stellt einen zusätzlichen Schutzmechanismus dar, der es erlaubt, korrupte oder ungeprüfte Dateisysteme ohne Gefahr für das Basissystem zu überprüfen.

Bisher wurde durch den Begriff "Dateiobjekt" impliziert, daß die Dateidaten als durch das übergeordnete Dateisystem verwaltete und strukturierte Einheit betrachtet werden. Es muß aber darauf hingewiesen werden, daß das Dateisystem als Ganzes und damit die Gesamtheit der Dateidaten der untergeordneten Dateiobjekte über die Gerätedatei des logischen Plattenlaufwerks ebenfalls als Dateiobjekt ansprechbar und mit Zugriffsrechten versehen ist.

Mechanismus "*Discretionary Access Control*" für IPC-Objekte:

IPC-Objekte werden über ihren Namen (*key*) bzw. die beim Erzeugen eines solchen Objekts vom System vergebene Kurzkennung identifiziert.

Die Zugriffsrechte auf die IPC-Objekte "Message Queue" und "Semaphore" werden im *Permission Mode Word* (PMW) des Objekts in einer Verwaltungsstruktur im Hauptspeicher gehalten. Zugriffsrechte können getrennt vergeben werden für den Erzeuger bzw. den aktuellen Eigentümer des Objekts, die Erzeuger-Gruppe bzw. Eigentümer-Gruppe des Objekts und alle anderen Benutzer. Es wird unterschieden zwischen Lese- und Schreibrecht. Diese Rechte betreffen gleichermaßen die Verwaltungs- und die Nutzdaten der IPC-Objekte.

Der Zugriff auf den *key* (Namen) oder die Kurzkennung eines IPC-Objekts erfordert keine Berechtigung.
Die Zugriffsrechte eines Subjekts an einem IPC-Objekt sind gegeben

- durch seine Identität (das sind die effektive Benutzer-Kurzkennung und die effektive Gruppen-Kurzkennung des zugreifenden Prozesses) und
- durch die Identität (des Erzeugers, Eigentümers und der entsprechenden Gruppen) und das PMW des Objekts.

Die Zugriffsrechte im PMW werden beim Anlegen des Objekts durch den Eigentümer gesetzt; dies erfolgt über Systemaufrufe, die von System-Kommandos (Kommandos, die vom Hersteller mit dem System ausgeliefert werden) oder Anwenderprogrammen aufgerufen werden. Beim Anlegen eines IPC-Objekts werden in der zugehörigen Kontrollstruktur die Eigentümer- und Erzeuger-Kurzkennung und die zugehörigen Gruppen-Kurzkennungen mit den entsprechenden effektiven Kurzkennungen des Erzeugerprozesses belegt.
Die Zugriffsrechte im PMW können nachträglich durch den Eigentümer, den Erzeuger oder den *Super User* geändert werden. Zur Änderung der aktuellen Benutzer-Kurzkennung oder Gruppen-Kurzkennung sind nur der Erzeuger des IPC-Objektes und der *Super User* berechtigt.

Der Mechanismus erfaßt die IPC-Objekte "Messages Queue" und "Semaphore" und damit alle IPC-Objekte, die in den Sicherheitsanforderungen als der Rechteverwaltung unterliegende IPC-Objekte benannt sind.

Eine indirekte Rechteänderung durch Änderung des Eigentümers oder der Gruppenidentität des Objekts kann nur durch den Erzeuger selbst geschehen. Da die Systemaufrufe, mit denen Rechte an IPC-Objekten verändert werden, von jedem Programm benutzt werden können, ist eine verdeckte Rechteänderung über "Trojanische Pferde" möglich; dazu ist allerdings erforderlich, daß das Programm von einem zu der Rechteveränderung berechtigten Benutzer ausgeführt wird.
Ein Wechsel der Erzeuger-Identität ist nicht möglich.
Die Gefährdung, die von einer verdeckten Rechteänderung im Fall von IPC-Objekten ausgeht, ist geringer als bei Dateien, da IPC-Objekte in der Regel nur temporär im System existieren.

Mechanismus des sog. "*sticky bit*" zur Einschränkung des Löschens von Dateien in Verzeichnissen:

Der Mechanismus stellt eine Ergänzung zur Rechteverwaltung für Dateiverzeichnisse und Dateien dar und wirkt in Verbindung mit dem Schreibrecht für Verzeichnisse und Dateien. Die durch das DAC gegebenen Rechteverhältnisse werden lediglich für einen speziellen Fall (Schreibrecht für Verzeichnis, kein Eigentum an der Datei) eingeschränkt.

Das sog. *sticky bit*, das im PMW eines Dateiverzeichnisses global für alle Dateien gesetzt werden kann, verbietet einem Subjekt das Löschen von Dateien aus einem Verzeichnis, für das es das Schreibrecht besitzt, sofern es sich nicht um den Eigentümer der Datei oder den *Super User* handelt. Ist das *sticky bit* im Verzeichnis nicht gesetzt, so ist ein Löschen von Dateien zulässig.
Die Vergabe des *sticky bit* erfolgt nur durch den *Super User*; der Eigentümer eines Verzeichnisses kann das Bit löschen.

SUID / SGID-Mechanismen und zugehörige Schutzmechanismen:

Der SUID / SGID-Mechanismus ist im eigentlichen Sinn kein Mechanismus, der Sicherheitsanforderungen realisiert.
Er ist aus der Notwendigkeit heraus begründet, daß das System für die Erledigung bestimmter Aufgaben Kommandos benötigt, deren Zugriffsmöglichkeiten größer sind als (in der Regel) die Rechte des aufrufenden Benutzers. Der SUID / SGID-Mechanismus bietet die Möglichkeit, normale Benutzer temporär mit zusätzlichen Rechten auszustatten. Dies bedeutet de facto eine Einschränkung der Sicherheitsmechanismen, muß aber vor dem Hintergrund der Alternative gesehen werden, entweder entscheidende Funktionalität einzubüßen oder bestimmte Prozesse mit höheren Privilegien zu versehen.

Es handelt sich hierbei um zwei Mechanismen, die es ermöglichen, die für die Rechteprüfung relevanten Benutzer- und Gruppen-Kurzkennung zu verändern:

- Das s-Bit im PMW von ausführbaren Dateien bewirkt, daß die effektive Benutzer-Kurzkennung bzw. Gruppen-Kurzkennung des Prozesses bei der Ausführung auf die Kurzkennung des Eigentümers bzw. der Gruppe der Programmdatei gesetzt wird. Das s-Bit kann sowohl vom *Super User* als

auch vom Dateieigentümer vergeben werden. Letzterer benötigt dafür das Systemprivileg *chmodsugid*.

- Mit dem Systemaufruf *setuid* ist es dem *Super User* möglich, die effektive und die reale Benutzer-Kurzkennung auf einen beliebigen Wert zu setzen. Einem normalen Benutzer ist es lediglich möglich, die effektive Benutzer-Kurzkennung zurück auf die reale / gesicherte Benutzer-Kurzkennung zu setzen. Der Systemaufruf *setgid* bewirkt die entsprechenden Änderungen der Gruppen-Kurzkennung.

Um die Gefahr von Sicherheitslücken durch Ausnutzung des SUID / SGID-Mechanismus zu verringern, sind spezielle Schutzmechanismen implementiert worden:

- Beim Kopieren von SUID / SGID-Dateien oder beim modifizierenden Zugriff auf solche werden die s-Bits der Datei gelöscht.
- Das Setzen des s-Bits einer Datei ist nur Eigentümern mit speziellem Systemprivileg *chmodsugid* möglich; nur der *Super User* kann s-Bits an Dateien, deren Eigentümer er nicht ist, setzen.
- Das "Verschenken" einer Datei ist nur Benutzern mit speziellem Systemprivileg *chown* möglich; dabei werden die s-Bits gelöscht und können nur vom *Super User* oder dem neuen Eigentümer, sofern er das *chmodsugid*-Privileg hat, wieder gesetzt werden.
- Das Ausführen von SUID / SGID-Programmen ist nur Benutzern mit speziellem Systemprivileg *execsuid* möglich.
- Systemkommandos mit SUID / SGID-Eigenschaften, insbesondere solche mit SUID für *root*, wurden soweit wie möglich reduziert und in Bezug auf Sicherheitslücken überprüft.
- Der *Super User* ist angewiesen, SUID / SGID-Programme in Bezug auf Existenz und Änderung regelmäßig mit dem Sollzustand zu vergleichen. Für diesen Zweck existieren spezielle Dienstprogramme.

Diese Schutzmechanismen gegen die SUID / SGID-Mechanismen sind dazu geeignet, ein Einbringen weiterer SUID / SGID-Programme ins System zu unterbinden bzw. sie unter Kontrolle des *Super User*s zu stellen.

Die Überschaubarkeit der Rechteverwaltung ist durch den SUID / SGID-Mechanismus erheblich eingeschränkt, da einem normalen Benutzer nicht

erkennbar ist, welche anderen Benutzer auf diesem Wege auf ihm gehörende Objekte zugreifen können. Es sind keine ausgleichenden Mechanismen vorhanden, die die Überschaubarkeit in Bezug auf SUID / SGID-Mechanismen verbessern.

SUID / SGID-Programme mit privilegiertem Eigentümer können verdeckte Rechteänderungen bewirken, wenn ihr Code entsprechende Systemaufrufe enthält. Ein solches Programm kann von einem Benutzer nur dann gezielt zur Umgehung der Sicherheitsfunktionen ausgenutzt werden, wenn die Rechte an einem gewünschten Objekt in der Weise geändert werden, daß unbefugte Benutzer (er selbst oder andere) die Zugriffsberechtigung erlangen oder der Zugriff für berechtigte Benutzer gesperrt wird. Eine solche Rechteveränderung an einem gewünschten Objekt ist dann möglich, wenn

- dieses Objekt mit seinem Pfadnamen im Quellcode des Programms als Objekt der Rechteveränderung genannt ist,
- dieses Objekt mit seinem Pfadnamen in einer *include*-Datei steht und dadurch in den Programmcode eingeht,
- der Name vom Programm zur Laufzeit aus einer Datei gelesen wird oder
- der Pfadname dem Programm ganz oder teilweise als Argument übergeben wird.

Es ist sicherzustellen, daß SUID / SGID-Programme der letzten Art ins System nicht eingebracht werden und daß Dateien, die Input-Funktion für SUID / SGID-Programme haben, nicht durch Unbefugte verändert werden können. *Include*-Dateien dürfen nicht schlechter geschützt sein als die anderen Quelldateien des Programms.

Es muß verhindert werden, daß Benutzer unkontrolliert eigene SUID / SGID-Programme in das System einbringen können, indem z.B. der *Super User* Programme aus einem schlecht geschützten Verzeichnis übernimmt und mit entsprechendem Eigentümer und s-Bit versieht.

Für das Löschen / Sperren von Eigentümern / Gruppen von SUID / SGID-Programmen gilt das gleiche wie im allgemeinen Fall. Es kommt in diesem Fall aber dazu, daß auch nach dem Ausloggen dieses Benutzers seine Zugriffsrechte genutzt werden können, indem andere Benutzer durch Ausführung seiner SUID / SGID-

Programme seine effektive Benutzeridentität erlangen können, wenn auch nur in dem durch die Programme vorgegebenen Rahmen.

Implizite Rechteverwaltung für den Zugriff auf Prozesse:

Prozesse als Objekte unterliegen **keiner** expliziten Rechteverwaltung. Sie werden identifiziert über die Prozeß-ID, die nicht änderbar ist.

Für den Zugriff auf Prozesse gelten allerdings bestimmte Regeln und Restriktionen, die durch das System verwaltet bzw. geprüft werden. Über die Systemaufrufschnittstelle hat der Benutzer Möglichkeiten, auf einzelne Variablen der Prozeßstrukturen zuzugreifen und sie in gewissen Grenzen zu modifizieren, wodurch sich wiederum Auswirkungen auf das Zusammenwirken mehrerer Prozesse ergeben. In dieser Weise ist eine Form der impliziten Rechteverwaltung gegeben.
Als Zugriff auf einen Prozeß wird der Zugriff auf seine elementaren Prozeßstrukturen gewertet, durch die Prozeßeigenschaften und die für die Interaktion zwischen Prozessen relevanten Identitäten gegeben sind. Dabei kann zwischen lesenden und modifizierenden Zugriffen unterschieden werden.
Für die meisten Zugriffe gilt, daß sie nur über einen Systemaufruf aus dem Prozeß selbst heraus erfolgen können, Subjekt- und Objekt-Prozeß also identisch sind; dies wird durch die Syntax der betreffenden Systemaufrufe gewährleistet. Für lesende Zugriffe gelten dabei keine Einschränkungen; modifizierende Zugriffe unterliegen in Abhängigkeit von Aufruf und effektiver Benutzer-Kurzkennung des aufrufenden Prozesses i.d.R. Beschränkungen bzgl. der Werte, die gesetzt werden können.

Die einzige Möglichkeit für normale Benutzer, modifizierend auf einen anderen als den eigenen Prozeß zuzugreifen, ist das "Senden" eines Signals. Mit den Systemaufrufen *kill* und *killpg* kann ein gewähltes Signal gezielt an einen Prozeß bzw. eine Prozeßgruppe "gesendet" werden. Andere Systemaufrufe können zur Folge haben, daß ein bestimmtes Signal an den eigenen, den Vaterprozeß oder an die Prozeßgruppe gesendet wird.
Das Recht, durch *kill* oder *killpg* auf einen anderen Prozeß (Ausnahme: *special processes*, s.u.) durch das Senden eines Signals zuzugreifen, ist allein gegeben

durch die effektiven und realen Benutzer-Kurzkennungen des sendenden und des empfangenden Prozesses; eine der beiden folgenden Bedingungen muß zutreffen:

- Die effektive Benutzer-Kurzkennung des sendenden Prozesses ist die des *Super Users*.
- Die effektive oder reale Benutzer-Kurzkennung des Sender-Prozesses ist gleich der effektiven oder realen Benutzer-Kurzkennung des Empfänger-Prozesses.

Zusätzlich muß gelten:

- Die Prozeß-ID des Empfänger-Prozesses besagt, daß es sich nicht um einen *special process* handelt.

Üblicherweise wird ein Prozeß als Reaktion auf einen Signalempfang beendet. Im Programmcode eines Prozesses kann aber veranlaßt werden, daß bestimmte Signalarten ignoriert werden oder daß beim Empfang eines bestimmten Signals spezieller Code ausgeführt wird und danach das Programm beendet oder dort weitergeführt wird, wo die Unterbrechung stattgefunden hat; dies ist allerdings nicht für alle Signalarten möglich. Für ein solches "Abfangen" oder Ignorieren von Signalen sind keine speziellen Rechte des Prozesses gefordert; es muß sich lediglich um einen für ein solches Vorgehen zulässigen Signaltyp handeln.

Sowohl das "Senden" als auch das "Abfangen" eines Signals stellt formal einen modifizierenden Zugriff auf den Prozeß dar, da beide Aktionen einen Eintrag in der Prozeßstruktur des Empfänger-Prozesses zur Folge haben.
Da die Prozeßstrukturen im Arbeitsspeicher stehen, ist ein Zugriff möglich. Dieser Zugriff wird durch die Mechanismen der *Discretionary Access Control* für Dateiobjekte kontrolliert und soll nur dem *Super User* oder indirekt über das zugehörige Subsystem Benutzern mit dem *mem*-Privileg für Lesezugriffe zugänglich sein. Letztere werden über die Kommandos des Subsystems (*ps, ipcs*) realisiert.

Mechanismus der Subsysteme zur Sicherung von Sicherheitsfunktionen:

Subsysteme schützen einzelne sicherheitsrelevante Teilbereiche des Systems. Es handelt sich dabei schwerpunktmäßig nicht um einen Mechanismus der Rechteverwaltung, sondern der Mechanismus dient im wesentlichen der Sicherung von

anderen Sicherheitsfunktionen des Systems (Identifizierung und Authentisierung, Beweissicherung).

Der Zugriff auf Programme und Dateien eines Subsystems ist nur mit einer dem Subsystem zugehörigen effektiven Gruppen-Kurzkennung möglich.[10)] Diese Kurzkennung kann durch die Ausführung spezieller bzgl. ihrer Sicherheitseigenschaften überprüfter SGID-Programme erlangt werden. Zur Ausführung eines solchen Programms oder einzelner Teilfunktionen benötigt ein Benutzer das jeweilige Kommandoprivileg, dessen Vorhandensein für den Benutzer von dem Programm überprüft wird. Einige Subsystem-Kommandos (z.B. *authif*) überprüfen das Privileg nicht, wenn der Aufrufer die Gruppenidentität des Subsystems besitzt. Zusätzlich muß der Benutzer zum Ausführen des Programms berechtigt sein und die jeweils erforderlichen Systemprivilegien besitzen.

Die Erweiterung der vorhandenen Subsysteme (z.B. durch Einbeziehung weiterer Kommandos) durch den *Super User* ist möglich, da die Basismechanismen (DAC für Dateiobjekte, SGID-Mechanismus, Abfrage von Benutzereigenschaften in Programmen) benutzergesteuert sind. Es ist möglich, weitere Subsysteme einzurichten. Nicht möglich ist es, die Kommandoprivilegien zu erweitern.

Zu einigen Subsystemen existiert ein zugehöriger Subsystem-Administrator, dem anstelle einer Login-Shell als Benutzerprogramm eine Menü-Schnittstelle zur Verwaltung der dem Subsystem zugehörenden Systemfunktionen und -dateien zugewiesen ist. Diese administrativen Benutzer können als Gruppen-Kurzkennung die Gruppen-Kurzkennung des Subsystems besitzen.

Es ist nicht vorgesehen, aber möglich (auch über das Menüsystem), an normale Benutzer die Gruppen-Kurzkennung des Subsystems zu vergeben. Auf diese Weise würde der für das Subsystem entscheidende SGID-Mechanismus umgehbar gemacht.

10) Da es sich um DAC-Rechte handelt, sind sie als solche grundsätzlich auch änderbar. Damit würde aber der Subsystem-Mechanismus wirkungslos. Deshalb müssen die Eigentums- und Zugriffsrechte an den Objekten der Subsysteme so gesetzt sein, daß kein unprivilegierter Benutzer die Rechte verändern kann.

Mechanismus der Kommandoprivilegien:

Bei den Kommandoprivilegien handelt es sich um einen Mechanismus, der zusätzlich zu den allgemeinen Mechanismen der Rechteverwaltung wirkt und der nur speziell geschützte Objekte betrifft.

Kommandoprivilegien berechtigen einen Benutzer, Dienste von Subsystemen in Anspruch zu nehmen, z.B. zur Ausführung von SGID-Programmen des Subsystems. Die Vergabe von Kommandoprivilegien erfolgt durch den *Super User* oder *Authentication Administrator*. Sie werden in der durch den DAC- und Subsystem-Mechanismus geschützten Paßwort-Datenbasis eines Benutzers gehalten; zur Prüfung wird die LUID des Benutzerprozesses herangezogen.
Das System sieht eine feste Menge von Kommandoprivilegien mit festgelegter Bedeutung und zugeordneten Subsystemen vor; die Anzahl der Kommandoprivilegien ist nicht erweiterbar.

Durch Tests des Herstellers und der Evaluatoren wurde geprüft, ob alle Subsystemkommandos nur mit dem zugehörigen Kommandoprivileg ausgeführt werden können. Für das Privileg *auth* ergab der Test, daß der Zugriff auch ohne das Privileg, mit anderer Authorisierung (DAC) möglich ist.

Sowohl der *Super User* als auch der *Authentication Administrator* sind in der Lage, aus der Paßwort-Datenbasis alle *auth*-Privilegien einschließlich der eigenen zu löschen. Da auch der *Super User* zur Ausführung privilegiengeschützter Kommandos nur berechtigt ist, wenn er das entsprechende Privileg besitzt, ist es ihm u.U. auf dem von dem System vorgesehenen Weg nicht mehr möglich, Kommandoprivilegien zu vergeben oder zu löschen und damit wieder einen Administrator mit *auth*-Privileg einzusetzen.
D*er Super User* kann aber auf Grundlage der Basis-Rechteverwaltung die Schutzmechanismen des *auth*-Subsystems außer Kraft setzen und manuell einen neuen *Authentication Administrator* einsetzen.

Der Schutz von Objekten durch Kommandoprivilegien ist nur wirksam im Zusammenhang mit speziellen Eigentums- und DAC-Zugriffsrechten an diesen Objekten. Werden diese Rechte verändert, so werden die Kommandoprivilegien

funktionslos. Ein spezielles Dienstprogramm erleichtert es dem *Super User*, die Übereinstimmung mit dem Soll-Zustand zu überprüfen.

Mechanismus der Systemprivilegien:

Bei den Systemprivilegien handelt es sich um einen Mechanismus, der zusätzlich zu den allgemeinen Mechanismen der Rechteverwaltung wirkt und der nur speziell geschützte Objekte betrifft.

Systemprivilegien berechtigen einen Prozeß zur Ausführung von bestimmten Aktionen (wie z.B. Zugriff auf das *Audit Subsystem*, Start von SUID-Programmen, u.a.); ohne das zugehörige Privileg soll ein Prozeß nicht in der Lage sein, die entsprechende Aktion durchzuführen.
Das System sieht eine feste Menge von Systemprivilegien für festgelegte Aktionen vor; die Anzahl der Systemprivilegien ist nicht erweiterbar.

Die Vergabe der Privilegien an Prozesse erfolgt beim Hochfahren des Systems; dabei wird der *init*-Prozeß mit maximalen Privilegien ausgestattet. Die Weitergabe von Privilegien erfolgt durch Vererbung auf die erzeugten Prozesse. Die jeweiligen Programme entziehen sich selbst mit einem speziellen Systemaufruf diejenigen Privilegien, die sie nicht (mehr) benötigen; speziell werden vor dem Start einer Benutzer-Sitzung durch den vorgelagerten System-Prozeß alle Privilegien, die dem Benutzer nicht zugeordnet sind, entzogen. Systemprivilegien können nicht zu einem späteren Zeitpunkt vergeben oder wieder aktiviert werden.

In der geschützten Paßwort-Datenbasis eines Benutzers sind diejenigen System-privilegien verzeichnet, mit denen nach dem Login-Vorgang sein Benutzerprozeß gestartet wird. Der Eintrag in die Datenbasis (und damit die Vergabe von System-privilegien an den Benutzer) erfolgt durch den Systemadministrator mit Kommandoprivileg *auth*.

Mechanismus der "*Super User*"-Privilegien:

Bei dem *Super User*-Privileg handelt es sich um einen Mechanismus, der nur teilweise der Rechteverwaltung unterliegt; obwohl die Benutzer-Kennung des *Super Users* wie jede andere Kennung mit dem DAC-Mechanismus für Datei- und IPC-Objekte verwaltet werden kann, ist dies - ausgenommen die Vergabe des

Eigentümers *root* / *admin* für SUID-Programme - eine Scheinverwaltung, da sie für die effektiven Zugriffsrechte des *Super Users* ohne Belang sind.

Das *Super User*-Privileg ist durch die effektive Benutzer-Kurzkennung 0 gegeben; das entspricht der Kurzkennung der administrativen Benutzer *root* und *admin*. Die Vergabe dieses Privilegs erfolgt automatisch mit der Vergabe der effektiven Identität an den Prozeß beim Login oder bei der Ausführung von SUID-*to-root*-Programmen.
Ein Prozeß mit *Super User*-Privileg unterliegt keinen Beschränkungen durch die Zugriffskontrolle des DAC-Mechanismus für Datei- und IPC-Objekte (Ausnahmen sind dadurch gegeben, daß auch der *Super User* nur eine Datei als Programm ausführen kann, wenn mindestens ein Ausführungsrecht für diese Datei gegeben ist und auch er nur lesend auf Dateien eines Dateisystems zugreifen kann, das nur mit der Leseoption gemountet wurde).
Er ist zur Änderung der DAC-Rechte und Eigentumsverhältnisse an allen Objekten sowie zum Einbringen und Sperren / Löschen von Benutzern berechtigt.
Außerdem berechtigt dieses Privileg einen Prozeß, bestimmte Systemaufrufe auszuführen oder sie mit einem erweiterten Aktionsrahmen zu benutzen. Ohne dieses Privileg soll ein Prozeß nicht in der Lage sein, diese Systemaufrufe bzw. die entsprechende Aktion durchzuführen. Dies betrifft speziell Systemaufrufe, mit der die System-Konfiguration oder einzelne Systemvariable geändert werden können.
Mit dem *Super User*-Privileg verfügt der Prozeß automatisch über alle Systemprivilegien.

Einschränkungen der Rechte des *Super Users* sind nur oberhalb der Systemaufrufschnittstelle - auf Programmebene - möglich. So kann durch Entzug der Kommandoprivilegien der *Super User* an der Ausführung von Subsystemprogrammen gehindert werden; analog sind in beliebigen Anwenderprogrammen Prüfungen des aufrufenden Benutzers implementierbar, durch die auch der *Super User* von der Ausführung des Programms oder von bestimmten Aktionen ausgeschlossen werden kann.

Das SINIX-S-System sieht Rollen und Privilegien vor, über die der *Super User* nicht unmittelbar aufgrund seiner Identität, sondern nur im Rahmen der Rechte

vergabe im System verfügt. Diese Rollen und Privilegien stellen eine gewollte Einschränkungsmöglichkeit des *Super User*-Privilegs dar.
Es entsteht ein Widerspruch dadurch, daß die Vergabe und Prüfung dieser Privilegien auf Verwaltungsdateien basiert, auf die der *Super User* unbeschränkten Zugriff hat, so daß er in der Lage ist, sich jedes gewünschte Privileg zu beschaffen.

Unbegrenzt sind die Möglichkeiten, die einem Benutzer durch das *Super User*-Privileg gegeben sind; grundsätzlich kann er alles realisieren, was die vorhandenen technischen Ressourcen erlauben; die damit verbundene reale Gefährdung hängt ab von der Böswilligkeit oder Leichtfertigkeit des Benutzers, seinen Kenntnissen über das System (Administrationsebene, Dateien, Programme), seinen Programmierfähigkeiten und dem betriebenen Aufwand.

Mechanismus zum Einbringen und Löschen / Sperren von Benutzern:

Die Benutzer, ihre Kurzkennungen, Gruppenzugehörigkeit und Privilegien werden in der geschützten Paßwort-Datenbasis gehalten. Diese Daten sind Teil des *Authentication Subsystems*; nur Administratoren mit dem Kommandoprivileg *auth* besitzen Zugriffsrechte für die Kommandos des Subsystems und die Datenbasis. Ein Menüsystem regelt alle Aktionen, die mit dem Einrichten und Sperren / Entsperren von Benutzern verbunden sind. Das Menüsystem sieht allerdings nicht vor, daß ein Benutzer vollständig aus dem System gelöscht wird. Durch das Sperren wird der Benutzer am Zugang zum System (Identifikation und Authentisierung) gehindert. Die Rechte der ihm zugeordneten Kennungen sind davon nicht berührt.

Eine gewisse Unübersichtlichkeit entsteht dadurch, daß es mehrere Arten von Authorisierungen gibt, die zum Einbringen / Löschen / Sperren / Entsperren von Benutzern berechtigen, und daß dies, in Abhängigkeit von der jeweiligen Authorisationsart, auf verschiedenen Wegen erfolgen kann:

- Ein Benutzer mit *auth*-Kommandoprivileg ist berechtigt zum Ausführen der Kommandos des *Authentication Subsystems*, inkl. Menüsystem.
- Für den Benutzer mit *auth*-Benutzerkennung wird nach dem Login *das auth*-Menüsystem gestartet; eine andere Anwendung ist für diesen

Benutzer nicht vorgesehen, läßt sich aber mit dem *auth*-Menüsystem leicht verschaffen.
- Benutzer mit *auth*-Benutzer oder *auth*-Gruppenkennung sind berechtigt zum Ausführen der Kommandos des *auth*-Subsystems (bis zur Prüfung des Kommandoprivilegs) und zum direkten Zugriff auf die Dateien des Subsystems.
- Der *Super User* ist berechtigt zum beliebigen Zugriff auf alle Objekte des Systems; er unterliegt bei den entsprechenden Kommandos der Prüfung des Kommandoprivilegs.

Da für bestimmte Aktionen zusätzliche Rechte oder Systemprivilegien vorausgesetzt werden, sind die aufgezählten Authorisierungen nicht immer hinreichend.

Das System enthält keinen Schutz dagegen, daß von mehreren authorisierten Benutzern an verschiedenen Bildschirmen aus gleichzeitig die Paßwortdaten, ggf. auch desselben Benutzers, geändert werden können. Die mit dem Menüsystem auf dem Bildschirm sichtbaren und in der Datei vorhandenen Paßwortdaten sind dann nicht konsistent. Die erste Änderung wird dann durch die zweite rückgängig gemacht, ohne daß dieser Vorgang den beiden Benutzern transparent ist.

Einige Benutzereigenschaften, die dem Benutzer beim Eintragen ins System zugewiesen werden, können im Rahmen des dafür vorgesehenen Menüsystems später nicht mehr geändert werden; dazu gehören die Benutzerkennung, die Benutzer-Kurzkennung und das *HOME*-Verzeichnis. Diese Einschränkung bedeutet einen Schutz vor der Erzeugung ungewollter und unüberschaubarer Nebeneffekte für die Rechteverwaltung, die bei nachträglicher Änderung dieser Werte entstehen könnten. In Bezug auf das *HOME*-Verzeichnis besteht durch diese Einschränkung aber auch die Gefahr, daß der Administrator zur Revidierung irrtümlicher oder nicht mehr zweckmäßiger Zuweisungen die gewünschten Änderungen unter Umgehung des Menüsystems durchführt, so daß die Kontrolle durch das Menüsystems dann vollständig entfällt.

Der *Super User* kann durch manuelles Editieren der Authentisierungsdateien unbeabsichtigt einen inkonsistenten Zustand herstellen, der zur Folge hat, daß jeder Authentisierungsversuch (auch des *Super Users*) abgewiesen wird und das System nicht mehr zugänglich ist.

3.3.3 Qualität der Mechanismen zur Rechteprüfung

Mechanismus der Prüfung beim Zugriff auf Dateiobjekte:

Bei den Zugriffen auf Dateiobjekte ist zu unterscheiden zwischen Zugriffen, die über einen Datei-Identifikator (Deskriptor) erfolgen und eine geöffnete Datei voraussetzen, und solchen, die das Dateiobjekt über seinen Namen (Dateipfad) ansprechen. Während die Zugriffe der ersten Art in einer Zugriffssequenz erfolgen und dabei Zugriffe auf Teile des Dateiinhalts möglich sind, betreffen die Zugriffe über den Dateinamen das Objekt global, d.h. seine Verwaltungsstrukturen oder die Datei als Ganzes.

Die Prüfung des Zugriffs auf ein Dateiobjekt erfolgt beim Eröffnen der Datei in den Systemaufrufen *open* oder *creat*, die den Datei-Identifikator für weitere Zugriffe zur Verfügung stellen. In den Parametern des Aufrufs wird die gewünschte Zugriffsart angegeben. Bei der Prüfung des Zugriffs wird der vollständige Pfadname der Datei zugrunde gelegt.
Besitzt der aufrufende Prozeß nicht für alle Verzeichnisse des Pfads das Suchrecht oder sind im PMW des Objekts für die effektive Benutzer- oder Gruppen-Kurzkennung des Prozesses die erforderlichen Rechte nicht eingetragen, so wird der Systemaufruf abgebrochen und gibt anstelle des Datei-Identifikators einen Fehlercode zurück. Ist die effektive Benutzer-Kurzkennung 0 (*Super User*), ist der Zugriff in jedem Fall zulässig.
Die Entscheidungsdaten für die Prüfung liegen einerseits in der Verwaltungsstruktur (*vnode*) der Dateiobjekte im Dateisystem, andererseits in der Verwaltungsstruktur des aktiven Prozesses (*proc*).

Ein Lese- oder Schreibzugriff auf ein Dateiobjekt ist nur möglich, wenn zuvor ein *open* oder *creat* durchgeführt wurde, da für den Zugriff der Datei-Identifikator angegeben werden muß. Beim Zugriff wird im Systemaufruf geprüft, ob der angegebene Datei-Identifikator für den Prozeß und die Zugriffsart zulässig ist. Eine Prüfung der Berechtigung des Subjekts erfolgt nicht mehr. Wird versucht, auf das Dateiobjekt einen Zugriff auszuüben, der sich von der beim Eröffnen der Datei angegebenen Zugriffsart unterscheidet, so wird dieser mit einem Fehlercode abgewiesen. Die Prüfung erfolgt in den Systemaufrufen *read* bzw. *write* (bzw. *readv*, *writev*).

Die Entscheidungsdaten für diese Prüfung liegen einerseits in der Verwaltungsstruktur der geöffneten Datei (Datei-Deskriptor), andererseits in der Verwaltungsstruktur des aktiven Prozesses (*proc*).

Auf eine geöffnete Datei können folgende Operationen angewendet werden:

- *dup, dup2, fsync, lseek, fstat, close, fstatfs* (betr. Dateisystem), *ioctl* (für Gerätedateien), *fcntl, flock, fchown, fchmod, ftruncate, locking, mmap, tell, rdchk, select*

Die Berechtigung für die Ausführung einer Programmdatei wird im Systemaufruf *exec* (*execv* bzw. *execve*) geprüft; auch der *Super User* unterliegt in diesem Fall der Rechteprüfung. Er ist zur Ausführung nur berechtigt, wenn im PMW mindestens ein Ausführungsrecht gegeben ist.

Die Berechtigung für Zugriffe, die nur die Verwaltungsstruktur (*vnode*) eines Dateiobjekts betreffen, wird ebenfalls in den entsprechenden Systemaufrufen geprüft; hier wird i.d.R. auf die nicht geöffnete Datei mit Angabe des Pfadnamens zugegriffen (Ausnahmen: *fchmod*, *fchown*), wobei generell das Suchrecht für alle Verzeichnisse des Pfads vorausgesetzt und geprüft wird (dies gilt entsprechend, wenn der Systemaufruf mehrere Pfadnamen als Parameter vorsieht).

Verändernde Zugriffe sind:

- *chmod, chown, link, rename, unlink, mknod, mkdir, rmdir, utime, utimes, truncate, symlink, csymlink*

Veränderungen an einer Datei sind auch möglich nach *mmap*.

Lesende Zugriffe sind:

- *access, lstat, stat, statfs, ustat, readlink, readclink*

Für diese lesenden Zugriffe sind außer dem Suchrecht für die Pfad-Verzeichnisse keine Rechte an dem Dateiobjekt gefordert.

Weitere Zugriffe, die Dateien oder Dateisysteme betreffen, sind:

- *chdir, chroot, mount, getdirentries, umount, select, munmap*

Modifizierende Zugriffe auf Dateiobjekte werden mit einem Fehlercode zurückgewiesen, wenn sie sich in Dateisystemen befinden, zu denen nur Lesezugang besteht oder wenn es sich bei dem Objekt um eine Datei handelt, die gerade als Programm ausgeführt wird. Die Prüfung erfolgt in den jeweiligen Systemaufrufen.
Das Löschen eines Dateiobjekts aus einem Verzeichnis, dessen *sticky bit* gesetzt ist, wird mit einem Fehlercode zurückgewiesen, wenn das Subjekt nicht gleich dem Eigentümer des Objekts oder dem *Super User* ist; die Prüfung erfolgt im Systemaufruf *unlink*.
Zugriffe auf unzulässige Speicherbereiche werden durch die Mechanismen der Speicherverwaltung mit dem Signal *SIGSEGV* (teilweise auch *SIGBUS*) abgewiesen. Diese Prüfung erfaßt auch Zugriffe auf durch *mmap* zugänglich gemachte Dateiteile.

Ein Speichermedium für ein Dateisystem, z.B. ein logisches Plattenlaufwerk, stellt i.a. seinerseits als Gerätedatei ein DAC-geschütztes Objekt dar. Werden die Rechte an einer solchen Datei fälschlicherweise so gesetzt, daß andere Benutzer als der *Super User* Zugriffsrechte bekommen, so besteht die Möglichkeit des Zugriffes auf Dateiinhalte aller Dateien des Dateisystems unter Umgehung der an den einzelnen Dateiobjekten vergebenen DAC-Rechte.

Der Mechanismus prüft nicht jeden Zugriff auf ein Dateiobjekt dahingehend, ob der zugreifende Benutzer zum aktuellen Zeitpunkt zu diesem Zugriff berechtigt ist. Auf diese Weise ist es möglich, daß Zugriffe erfolgen, die zum aktuellen Zeitpunkt gegen die durch die Rechteverwaltung festgelegten Rechte verstoßen. Dies kann der Fall sein, wenn

- zwischen dem Anlegen bzw. Eröffnen der Datei und dem Zugriff dem Benutzer Rechte durch den *Super User* oder Eigentümer des Objekts entzogen wurden,
- sich zwischen dem Anlegen bzw. Öffnen der Datei und dem Zugriff die effektive Benutzer- oder Gruppen-Kurzkennung ändert oder
- der Benutzer zwischen dem Anlegen bzw. Eröffnen der Datei und dem Zugriff durch den *Super User* gesperrt wurde.

In allen diesen Fällen ist der Benutzer zum Zeitpunkt des Öffnens der Datei zu dem späteren Zugriff berechtigt.

Mechanismus der Prüfung beim Zugriff auf IPC-Objekte "Semaphore" und "Message Queue":

Die Berechtigung für Zugriffe auf die IPC-Objekte Semaphore und Message Queue wird in den jeweiligen Systemaufrufen geprüft. Ist das Subjekt zu dem Zugriff nicht berechtigt, so wird der Systemaufruf abgebrochen und ein Fehlercode zurückgegeben. In speziellen Fällen wird der Zugriffsversuch ignoriert, z.B. bei dem Versuch, den Erzeuger oder die Erzeugergruppe eines Objekts zu ändern.

Die Identifikation existierender IPC-Objekte erfolgt über einen numerischen Schlüssel (*key*) oder über eine Kurzkenung. Die Zuordnung der Kurzkennung zum Schlüssel erfolgt mit den Systemaufrufen *semget* oder *msgget*, es ist aber auch möglich, eine Kurzkennung für Zugriffe auf IPC-Objekte zu benutzen, ohne ihn vorher über *semget* oder *msgget* angefordert zu haben.

Beim Zugriff mit *semget* oder *msgget* auf ein existierendes IPC-Objekt dürfen die als Parameter übergebenen Rechte nicht über die im PMW des Objekts gesetzten Rechte hinausgehen.

Das Lesen der Eigentümer- und Gruppen-Kurzkennung und des PMWs mit *semctl* oder *msgctl* erfordert Lese- oder Eigentumsrechte. Zum Ändern des PMW mit *semctl* oder *msgctl* sind nur der Erzeuger, Eigentümer oder *Super User* berechtigt. Der Eigentümer kann nur durch den Erzeuger oder den *Super User* geändert werden; der Erzeuger benötigt dazu das *chown*-Privileg. Das gleiche gilt für das Ändern der Gruppenzugehörigkeit.
Das Lesen von Statusinformationen mit dem Kommando *ipcs* erfordert Leseberechtigung oder das Kommandoprivileg *mem*.
Bei jedem Lese- und Schreibzugriff auf ein IPC-Objekte wird die Berechtigung des Benutzers für den Zugriff geprüft:

- *msgsnd*, *semop* (Senden einer Nachricht, Ändern eines Semaphor) : Schreibrecht
- *msgrcv*, *semop* (Empfangen einer Nachricht, Ändern eines Semaphor) : Leserecht

Der Zugriff auf den *key* oder die Kurzkennung eines IPC-Objektes unterliegt keiner Prüfung.

Mechanismus der Prüfung beim Zugriff auf Prozesse:

Lesende Zugriffe auf einen Prozeß werden nicht geprüft, wenn sie über Systemaufrufe aus dem Prozeß selbst heraus erfolgen. Dies betrifft die Systemaufrufe:

- *getegid, geteuid, getgid, getgroups, getpgrp, getpid, getppid, getuid, getluid, getpriv, umask, ulimit, universe, times, getdtablesize, getitimer, getlimit, getpriority, getrusage*

Indirektes Lesen der Prozeßstrukturen über das Lesen des Arbeitsspeichers ist geschützt durch den DAC-Mechanismus und zusätzlich durch das erforderliche Privileg *mem*. Dies betrifft die Kommandos *ps* und *ipcs*.

Modifizierende Zugriffe auf die Prozeßstrukturen des eigenen Prozesses werden realisiert durch die Systemaufrufe:

- *setuid, setreuid, setgid, setregid, setgroups, setpgrp, setluid, setpriv, setsid, signal, umask, ulimit, universe, nice, setitimer, setrlimit, setpriority*

Für die Prüfung bei modifizierenden Zugriffen gilt im Prinzip das gleiche wie für lesende Zugriffe; in Abhängigkeit von den für den jeweiligen Aufruf möglichen Parametern finden Prüfungen bzgl. der Zulässigkeit der einzutragenden Werte, i.d.R. in Abhängigkeit von der effektiven Benutzer-Identität des aufrufenden Prozesses, statt. Um in eine andere Gruppe wechseln zu können, muß der Benutzer Mitglied dieser Gruppe oder *Super User* sein.

Ein modifizierender Zugriff auf einen anderen Prozeß stellt das "Senden" eines Signals dar. Die Berechtigung eines Prozesses zum Senden eines Signals an einen Prozeß wird geprüft in den Systemaufrufen *kill* und *killpg*; andere Aktionen oder Systemaufrufe, die ebenfalls dazu führen, daß an einen Prozeß ein Signal gesendet wird, können in der Regel nur unter Bedingungen auftreten, wo die erforderlichen Rechte per se gegeben sind. Bei der Ausführung im Kern wird geprüft, ob das Signal vom empfangenden Prozeß ignoriert werden soll; in diesem Fall wird der Zugriff nicht zurückgewiesen, aber auch nicht durchgeführt. Ist der Senderprozeß zum Senden des Signals nicht berechtigt, so wird der Systemaufruf abgebrochen und ein Fehlercode zurückgegeben.

Für das Abfangen oder Ignorieren von Signalen durch den Systemaufruf *signal* sind keine Rechte des Empfängerprozesses erforderlich - es erfolgt also keine Prüfung; es muß sich lediglich um ein für diese Aktion zulässiges Signal handeln.

Neben den Zugriffen, die einzelne Prozesse betreffen, bietet das System auch Möglichkeiten, Informationen über das Gesamtsystem abzufragen.
Das Ändern von einzelnen Systemvariablen ist in den Fällen, wo das System eine solche Möglichkeit vorsieht, nur dem *Super User* möglich.

Mechanismus der Prüfung beim Zugriff auf Subsysteme:

Bei den Subsystemen handelt es sich nicht um einen eigenständigen Mechanismus, sondern um eine Kombination von anderen Mechanismen des Systems, mit der ein besonderer Schutz für Teilbereiche erreicht werden soll. Das Konzept basiert auf der Kombination der Mechanismen:

- *Discretionary Access Control* für Dateiobjekte (DAC)
- SUID / SGID-Mechanismus (betrifft nicht die Rechteprüfung)
- Kommandoprivilegien
- Systemprivilegien

Die Prüfungen innerhalb eines Subsystems erfolgen im Rahmen dieser Mechanismen.

Ein nichtprivilegierter[11)] Benutzer kann nicht direkt auf die Objekte in Subsystemen zugreifen, sondern nur vermittelt über spezielle SGID-Programme. Er benötigt für die Ausführung dieser Programme neben der Berechtigung der DAC das zugehörige Kommandoprivileg, das in seiner geschützten Paßwort-Datei und in der Datenbasis des Subsystems hinterlegt sein muß. In den jeweiligen Programmen wird geprüft, ob der zur LUID des zugreifenden Prozesses gehörende Benutzername in der Datenbasis des Subsystems eingetragen ist.

Es ist möglich, Zugang zu den Objekten des Subsystems zu erlangen, ohne das zum Subsystem gehörende Kommandoprivileg zu besitzen. Diese Möglichkeit besteht allerdings nur für Benutzer mit privilegierter Gruppen- oder Benutzer-Kurzkennung.

11) Ein Benutzer, der durch den Administrator die entsprechende Gruppenzugehörigkeit zum Subsystem zugewiesen bekommt, ist in diesem Sinn auch als "privilegiert" anzusehen.

Die Prüfung des Kommandoprivilegs als erste Bedingung für die Ausführung eines Kommandos oder einer Kommandooption erfolgt nicht durchgängig. Es ist allerdings nicht erkennbar, daß dadurch unprivilegierte Benutzer (d.h. Benutzer, die weder über Kommandoprivilegien noch über privilegierte Identität oder Gruppenzugehörigkeit verfügen) Subsystemkommandos ausführen können.

Es ist möglich, daß auf eine Datei gleichzeitig mehrfach zugegriffen wird, daß z.B. auf die Paßwort-Datei geschrieben wird, während sie sich im Lesezugriff befindet, und somit die Integrität der Entscheidungsdaten aus Sicht des Benutzers nicht gegeben ist. Die Gefahr, daß eine solche Situation entsteht, wird dadurch stark reduziert, daß für jeden Benutzer eine eigene Paßwort-Datei existiert und daß nur ein Systemadministrator mit *auth*-Privileg schreibend auf diese Dateien zugreifen darf. Durch umsichtiges Vorgehen des *Authentication Administrators* läßt sich das Entstehen einer solchen Situation vermeiden.

Mechanismus der Prüfung beim Zugriff auf durch Kommandoprivilegien geschützte Objekte:

Kommandoprivilegien berechtigen zum Zugang zu Subsystemen; die SGID-Programme, die den Zugang zu den Objekten der Subsysteme vermitteln, prüfen, ob der aufrufende Prozeß das erforderliche Kommandoprivileg besitzt. Dies geschieht, indem aus der allgemein lesbaren Paßwort-Datei der zur LUID des Prozesses gehörende Benutzername ermittelt wird und dann geprüft wird, ob dieser Name in der geschützten Datenbasis des Subsystems eingetragen ist.
Bei negativem Ergebnis der Prüfung hängt die Reaktion vom jeweiligen Subsystem und Kommando ab. Entweder das Kommando wird mit einer Fehlermeldung abgebrochen, oder die Ausführung des Kommandos wird auf die für nichtprivilegierte Benutzer vorgesehenen Aktionen eingeschränkt.
Unberechtigte Zugriffsversuche werden - abhängig von der Generierung - vom *Audit*-System protokolliert.

Ein Abgleich der Zuordnung Benutzer / Privileg in der Datenbasis des Subsystems und der Paßwortdatei des Benutzers erfolgt bei einer Privilegienänderung des Benutzers mit dem Menüsystem. Eine Inkonsistenz zwischen diesen Dateien ist möglich (sie entsteht z.B. dann, wenn ein Benutzer aus der */etc/passwd*-Datei gelöscht und dann mit dem Menüsystem mit anderen Privilegien wieder eingebracht wird). In diesem Fall stimmen die Entscheidungsdaten nicht mit den über das Menüsystem feststellbaren Privilegien des Benutzers überein.

Mechanismus der Prüfung beim Zugriff auf durch Systemprivilegien geschützte Objekte:

Der Zugriff auf durch Systemprivilegien geschützte Objekte erfolgt durch die jeweiligen Systemaufrufe (*exec*, *chmod*, *chow*n, Zugriffe auf */dev/auditw*). In diesen wird geprüft, ob es sich um einen Zugriff handelt, für den Systemprivilegien erforderlich sind, und - ggf. - ob der Prozeß die erforderlichen Systemprivilegien besitzt. Die Systemprivilegien werden im Authorisierungs-Vektor des Prozesses gehalten. Ist das Privileg nicht gegeben, so gibt der Aufruf einen Fehlercode zurück und wird beendet.
Zur Prüfung der vorhandenen Systemprivilegien sind der Systemaufruf *getpriv* und das Kommando *privs* vorhanden.
Der *Super User* ist von der Prüfung der an normale Benutzer vergebbaren Privilegien *chown*, *execsuid* und *chmodsugid* ausgenommen.

Mechanismus zur Prüfung der Zugriffe auf Objekte, die *Super User*-Identität erfordern:

Der Zugriff erfolgt durch die jeweiligen Systemaufrufe. In diesen wird geprüft, ob es sich um einen Zugriff handelt, für den *Super User*-Identität erforderlich oder hinreichend ist, und - ggf. - ob der Prozeß diese Identität besitzt. Ist die Identität nicht gegeben, so gibt im ersten Fall der Aufruf einen Fehlercode zurück und wird beendet; im zweiten Fall wird geprüft, ob der Prozeß aufgrund seiner effektiven Identität zu der Aktion berechtigt ist.

Systemaufrufe, zu denen nur der *Super User* berechtigt ist, sind:

- *acct*, *chroot*, *mknod* (außer Named Pipes), *mount*, *umount*, *reboot*, *setdomainname*, *sethostid*, *settimeofday*, *stime*, *quotacl*, *getfh*, *sethostname*, *setgroups*, *vhangup*, *adjtime*, *setluid* (nur initial)

Systemaufrufe, die nur vom *Super User* zur Erweiterung des gegebenen Werts genutzt werden können:

- *ulimit, setrlimit, setpriority, nice, setgroups*

Weitere Systemaufrufe, die dem *Super User* spezielle Möglichkeiten bieten:

- *kill, killpg, setpgrp, setregid, setreuid, setuid, setgid*

Mechanismus der Prüfung der Berechtigung zum Einbringen oder Löschen / Sperren von Benutzern:

Für das Einbringen und Sperren von Benutzern ist ein Menüsystem (*authif*) implementiert. Das Programm *authif*, weitere Kommandos und die Dateien, in denen die Benutzer-Informationen gehalten werden, bilden das *auth*-Subsystem. Die Kommandos sind SGID-*to*-*auth*-Programme und erfordern das *auth*-Kommandoprivileg. Die Dateien haben die *auth*-Gruppenzugehörigkeit und sind von dieser Gruppe zugreifbar.
Im System ist ein administrativer Benutzer mit Benutzernamen und Gruppe *auth* eingerichtet, dem als Login-Shell das Programm *authif* zugewiesen ist. Das Einrichten und Sperren von Benutzern durch den *Authentication Administrator* mit *authif* ist der vom System vorgesehene und empfohlene Weg.

Alternativ bestehen Zugriffsmöglichkeiten durch andere Benutzer mit *auth*-Kommandoprivileg sowie durch Benutzer mit privilegierter Identität wie *root* oder *auth*-Gruppenidentität (der Administrator ist angewiesen, keinem normalen Benutzer diese Gruppenidentität zuzuordnen). Hier ist zu unterscheiden zwischen

- dem ausführenden Zugriff auf die Programme, für die einerseits das DAC-Recht zum Ausführen erforderlich ist und für die andererseits, da es sich um SGID-Programme handelt, zusätzlich das Systemprivileg *execsuid* vorhanden sein muß, und bei denen durch das Programm selbst das Kommandoprivileg geprüft wird und
- dem schreibenden oder lesenden Zugriff auf die Programm- und Verwaltungsdateien.

Es ist vorgesehen, daß letzterer vermittelt über die Subsystem-Programme erfolgt, die Prüfung als solche erfolgt aber im Rahmen der DAC-Mechanismen und erfordert deshalb nur die Identität von *auth*-Gruppe oder *Super User*.

Die Prüfung der erforderlichen Privilegien für das Einbringen und Löschen / Sperren von Benutzern erfolgt also in Abhängigkeit davon, auf welchem Weg auf die Administrationsprogramme und -dateien zugegriffen wird.

Zum Problem der IPC-Objekte "Sockets":

Es gibt zwei Ausprägungen vom Typ Socket, zum einen *UNIX domain sockets* und zum anderen *Internet domain sockets*. Sockets realisieren abstrakte Kommunikationskanäle, von denen Nachrichten gesendet und empfangen werden.
UNIX domain sockets können mit dem Systemaufruf *bind* an einen Knoten im Dateisystem gebunden werden und sind dann von anderen Prozessen über diesen Namen (entspricht dem Dateinamen) bzw. Pfad ansprechbar.
Internet domain sockets sind speziell für die Kommunikation auf Basis der Protokolle TCP/IP bzw. UDP/IP zu Prozessen auf entfernten Systemen entworfen und werden über die sog. Portnummer, die im Kern verwaltet wird, identifiziert.

Da Sockets genau wie Dateiobjekte über Deskriptoren angesprochen werden, aber nicht der Rechteverwaltung unterliegen, müssen die Zugriffsmechanismen auf Dateiobjekte und Sockets gegeneinander abgegrenzt sein.

Der Zugriff auf Sockets erfolgt mittels spezifischer Systemaufrufe sowie durch Systemaufrufe, die auch auf Dateiobjekte anwendbar sind (*close*, *read*, *write*, *chmod*, *select*). Der Eintrag eines *UNIX domain sockets* im Dateisystem wird durch *unlink* gelöscht. Bei socketspezifischen Aufrufen wird geprüft, ob es sich bei dem Objekt um einen Socket handelt. Der Systemkern verfügt über die Information, ob es sich bei einem Knoten bzw. Deskriptor um einen Socket oder ein Dateiobjekt handelt. Er stellt durch Überprüfung dieser Eigenschaft in den Systemaufrufen sicher, daß die Schnittstellen von Datei- und Socketverwaltung nicht zu einer Umgehung der Sicherheitsfunktionen genutzt werden können. Repräsentiert der angesprochene Deskriptor keinen Socket, so wird der Zugriff mit einem Fehlercode abgewiesen. Wird im umgekehrten Fall ein Socket mit einem Systemaufruf angesprochen, der nur für Dateiobjekte zulässig ist (z.B. *open*), wird dieser Zugriffsversuch ebenfalls abgewiesen.

Ein Datenaustausch kann nur von Socket zu Socket erfolgen und erfordert die Identifizierung des Zielsockets mit seinem Namen bzw. seiner Portnummer. Socket-IPC basiert auf einer gesonderten Puffer-Verwaltung.
Der Versuch einer Weitergabe von Deskriptoren mit dem Systemaufruf *sendmsg* wird vom System abgewiesen.
Alle Prüfungen finden in den jeweiligen Systemaufrufen statt.

3.3.4 Mechanismus zur Beweissicherung

Die Beweissicherung erfolgt zum einen auf der Ebene der Systemaufrufe, d.h. die Benutzung der Systemschnittstelle durch Prozesse wird protokolliert, zum anderen generieren bestimmte vertrauenswürdige Systemprogramme (z.B. *login*), die dazu authorisiert sind, ihre Protokollsätze selbst.
Daneben können Subsystem-Prozesse Protokollsätze erzeugen, indem sie Nachrichten über eine *named pipe* an den sog. *audit daemon*, einen Prozeß außerhalb des Kerns, schicken.
Im Falle der Systemaufruf-Protokollierung wird nach der Rückkehr von einem Systemaufruf noch innerhalb des Kerns (in der *trap*-Routine) anhand verschiedener Kriterien geprüft, ob dieser Systemaufruf protokolliert werden muß oder nicht.
Diese Kriterien sind:

- Ob das *Audit Subsystem* eingeschaltet ist.
- Ob es sich um einen Systemaufruf handelt, der immer protokolliert werden muß (ein sog. "*mandatory system call*").
- Ob es sich um eine Applikation handelt, die ihre eigenen Protokollsätze schreiben darf (dann unterbleibt eine Protokollierung der in dieser Applikation auftretenden Systemaufrufe).
- Ob für den Benutzer / die Gruppe Systemaufrufe protokolliert werden sollen (vom *Audit Administrator* über *auditif* einstellbar).
- Ob der Systemaufruf in eine zu protokollierende Kategorie fällt (systemweit vom *Audit Administrator* einstellbar, benutzerspezifisch vom *Authentication Administrator* einstellbar).

Ergibt die Prüfung, daß der Systemaufruf zu protokollieren ist, so wird ein dem Systemaufruf und dessen Ergebnis entsprechender Protokollsatz erzeugt und direkt in eine geschützte Datei (ein sog. *collection file*) geschrieben. Eine Ausnahme bildet der Systemaufruf *exit*. Da dieser nicht zur *trap*-Routine zurück

kehrt, wird nach Prüfung der obigen Kriterien gegebenenfalls innerhalb der *exit*-Routine ein Protokollsatz geschrieben.

Im Falle der vertrauenswürdigen Systemprogramme schreibt die Applikation selbst Protokollsätze über das Device */dev/auditw* in das *collection file*, vorausgesetzt, das *Audit Subsystem* ist eingeschaltet und das entsprechende Ereignis (z.B. Login) fällt in eine Kategorie, die aufgezeichnet werden soll. Innerhalb des Device-Treibers wird geprüft, ob die Applikation zum Schreiben eigener Protokollsätze berechtigt ist. Dazu ist neben dem Schreibrecht auf */dev/auditw* die Autorisierung *writeaudit* notwendig.

Im Falle der Subsystem-Prozesse, die nicht über die *writeaudit* Autorisierung verfügen, schickt das Subsystem mittels *dlvr_audit* über eine geschützte *named pipe* den eigentlichen Protokollsatz an den *audit daemon*. Anhand des Gruppennamens des Aufrufers kann *dlvr_audit* verifizieren, daß der Protokollsatz von einem der Subsysteme stammt. Der *audit daemon* übernimmt dann den Protokollsatz, vorausgesetzt, das *Audit Subsystem* ist eingeschaltet und das entsprechende Ereignis fällt in eine Kategorie, die aufgezeichnet werden soll.

In jedem der drei beschriebenen Fälle zur Erzeugung von Protokollsätzen liest der *audit daemon* Protokollsätze (in den beiden ersten Fällen über das Device */dev/auditr* aus dem *collection file*, in letzterem Fall aus der *named pipe*), komprimiert sie und schreibt sie in eine geschützte Datei (*compaction file*).
Bei der Auswertung der Protokollsätze (der sogenannten *reduction*) durch den *Audit Administrator* mit Hilfe des Programms *auditif* werden die Protokollsätze aus der komprimierten Datei nach angebbaren Kriterien selektiert und in eine lesbare Form gebracht. Da die Protokollsätze das aktuelle Arbeitsverzeichnis und die LUID des verursachenden Benutzers selbst nicht enthalten, müssen dabei in einer Art Emulation die Erzeugung von Prozessen bzw. der Wechsel von Verzeichnissen nachvollzogen werden. Damit dies möglich ist, müssen manche Systemaufrufe (wie z.B. *exec* oder *chdir*) unbedingt protokolliert werden (das sind die "*mandatory system calls*").

Dieser Mechanismus stützt sich im wesentlichen auf Mechanismen zur Rechteprüfung und -verwaltung ab.

Eine Schwäche des Mechanismus besteht darin, daß zur Auswertung der Beweissicherung die ganze Historie des Ablaufs nachvollzogen werden muß, weil im Protokollsatz selbst die LUID und die absoluten Pfadnamen beteiligter Objekte nicht abgespeichert werden. Dadurch ist der Schaden, der durch Verlust auch nur eines Protokollsatzes entsteht, u.U. nicht mehr lokal begrenzt. Außerdem hat diese Schwäche zur Folge, daß sich, wenn das Beweissicherungs-System erst bei laufendem System eingeschaltet wird, nur für diejenigen Protokollsätze eine LUID rekonstruieren läßt, die von Benutzern erzeugt werden, die sich nach dem Zeitpunkt des Einschaltens angemeldet haben.

3.3.5 Qualität der Mechanismen zur Wiederaufbereitung

Mechanismen zur Wiederaufbereitung von Dateien auf Plattenspeicher:

Beim Löschen einer Datei mit dem Kommando *destroy* werden die Datenblöcke der Datei vor der logischen Freigabe physikalisch überschrieben.
Beim Löschen einer Datei mit dem Kommando *rm* werden die zugehörigen Plattenblöcke logisch freigegeben, deren Inhalt wird aber nicht physikalisch gelöscht. Durch Mechanismen bei der Dateihandhabung ist jedoch in diesem Fall sichergestellt, daß der nächste Benutzer, dem diese Datenblöcke für eine Datei zugeteilt werden, die darin enthaltene Information nicht lesen kann. Beschreibt der Benutzer die Datei lückenhaft, so werden die Lücken in den Datenblöcken mit "binär 0" aufgefüllt.
Der "normale" Benutzer kann nicht an die vorher in den Plattenspeicherblöcken enthaltenen Daten gelangen. Durch die Möglichkeit, unter Umgehung des Dateisystems direkt von der Platte zu lesen, kann der *Super User* auch Dateiinhalte von Dateien lesen, die mit *rm* gelöscht wurden.

Mechanismus zur Wiederaufbereitung des Arbeitsspeichers:

Bei SINIX-S V5.22 ist der Arbeitsspeicher in Speicherseiten aufgeteilt und wird virtuell adressiert. Dem Benutzer wird beim Erzeugen von Prozessen Arbeitsspeicher in Form von Speicherseiten zugeteilt. Die Wiederaufbereitung der Speicherseiten erfolgt vor der Zuteilung zu einem Prozeß (*fill on demand*), nicht bei der Freigabe durch einen Prozeß.
Neben dieser statischen Zuteilung beim Start eines Programms können weitere Speicherseiten dynamisch während der Laufzeit zugeteilt werden. In beiden Fällen

werden Speicherseiten vor der Zuteilung initialisiert (mit Programmcode, Werten initialisierter Variablen oder "binär 0").

Mechanismus bei der Wiederverwendung von Puffern bei der Interprozeß-kommunikation:

Solche Puffer werden für die Socket-IPC und Messages der Message-Queues verwendet.

Die Verwaltung der Puffer findet im Kern statt. Die Belegung und Freigabe der Einheiten erfolgt durch den Kern als Folge der die IPC-Objekte betreffenden Systemaufrufe. In beiden Fällen findet bei einer Wiederverwendung innerhalb des Kerns keine Wiederaufbereitung statt. Der Zugang der Benutzer zu den Speicher-inhalten erfolgt vermittelt über Systemaufrufe. Darunter befinden sich keine Operationen, die den Zugang zu nicht mehr gültigen bzw. noch nicht durch aktuelle Daten überschriebenen IPC-Puffern ermöglichen würden. Es werden beim Lesen von IPC-Objekten jeweils nur die Daten an den Benutzer übergeben, die im aktuellen Gültigkeitszyklus durch einen Schreibvorgang auf dieses Objekt erzeugt und noch nicht gelesen wurden. Die Länge des lesbaren Datenbereiches ist durch die Anzahl geschriebener Zeichen begrenzt. Ein Lesen über diesen Speicherbereich hinaus ist nicht möglich.

Mechanismus zur Wiederaufbereitung des Bildschirms der Datensichtstation:

Beendet ein Benutzer seine Arbeitssitzung (Logout), so wird der Bildschirm gelöscht und der "Begrüßungsbildschirm" ausgegeben, so daß für den nächsten Benutzer des Bildschirms keine Daten des Vorgängers mehr sichtbar sind.

Diese Art der Wiederaufbereitung stellt sicher, daß der nächste Benutzer des Terminals keine Informationen über die vorangegangene Sitzung erlangt. Voraus-setzung ist, daß jeder Benutzer seine Sitzung explizit beendet.
Es erfolgt kein automatisches Logout durch das System, wenn in einer Sitzung über einen längeren Zeitraum keine Aktion erfolgt.

Zur Wiederaufbereitung von Datenträgern (Diskette, Magnetband-Kassette):

Datenträger werden unter SINIX-S V5.22 über Gerätedateien angesprochen. Eine logische Zugriffssequenz besteht dabei aus dem Öffnen, Lesen / Schreiben und dem Schließen der Gerätedatei. Im Falle der Streamer-Gerätedatei (Magnetband-Kassette) kann zu einem bestimmten Zeitpunkt nur ein Benutzer diese logische Zugriffssequenz durchlaufen. Dadurch ist sichergestellt, daß nur dieser eine Benutzer während dieses Zeitraums auf das Band zugreifen, d.h. das eingelegte Band exklusiv benutzen kann. Im Falle der Disketten-Gerätedatei kann sich der Benutzer diesen exklusiven Zugriff nicht sichern.

Es existiert keine besondere Wiederaufbereitung für die Datenträger Magnetband-Kassette und Diskette. Entsprechende Maßnahmen (Entnahme der Magnetband-Kassette / Diskette nach dem gewünschten Lese- / Schreibvorgang) sind außerhalb des IT-Systems zu ergreifen.

3.4 Qualität der Abgrenzung zu nicht zu evaluierenden Systemteilen

Zu den sicherheitsrelevanten und damit zu evaluierenden Teilen eines Systems gehören alle Teile, die für die Einhaltung der Sicherheitspolitik und den Schutz der Objekte des Systems verantwortlich sind.
Bei dem Produkt SINIX-S V5.22 sind dies der Systemkern, die sicherheits-relevanten Programme und solche Dateien, die sicherheitsrelevante Daten enthalten.

Der Kern enthält den Code für die Basisfunktionen des Systems. Dazu gehören:

- Prozeßverwaltung und -scheduling
- Speicherverwaltung
- Ein- / Ausgabeverwaltung
- Dateiverwaltung
- Interprozeßkommunikation

Im Kern enthaltener Code zu Kommunikationszwecken ist nur im Zusammenhang mit spezieller Hardware aktivierbar, die nicht zum Konfigurierungsumfang des evaluierten Systems gehört (siehe 1.2).

Die Programmierschnittstelle wird durch Systemaufrufe angesprochen, deren Bearbeitung durch den Kern erfolgt.
Systemprogramme, die außerhalb des Kerns ablaufen, werden nur dann als sicherheitsrelevant betrachtet, wenn sie das SUID / SGID-Privileg besitzen und damit den SUID- / SGID-Mechanismus verwenden können.

Alle Dateien, die sicherheitsrelevante Informationen enthalten, sind Bestandteil der evaluierten Systemteile. Sie gruppieren sich in den fünf Unterverzeichnissen:
/dev, /etc, /tcb, /usr, /

Die Mechanismen zur Abgrenzung
- des Kerns,
- der Subsysteme,
- der übrigen SUID / SGID-Programme und
- der unbenutzten Treiber

können in ihrer Gesamtheit mit **stark** bewertet werden. Sie entsprechen somit den Anforderungen der Qualitätsstufe Q2.

3.4.1 Mechanismen zur Abgrenzung des Kerns

Es wird unterschieden zwischen Mechanismen, die sicherstellen, daß die genannten Schnittstellen nicht umgangen werden können, und solchen, die dafür sorgen, daß die genannten Schnittstellen ausschließlich in der dokumentierten Weise genutzt werden können und keine Umgehung der Sicherheitsfunktionen ermöglichen.

Mechanismen zum Schutz vor Umgehung der Schnittstelle:

Die Mechanismen bestehen in der
- Kapselung des Kerns durch Übergang in den Supervisor-Modus der CPU (Hardware-Status) bei Aufruf eines Systemaufrufs,
- Trennung der virtuellen Adreßräume von Kern und Benutzer-Prozessen über die MMU mit gleichzeitigem Hardware-Speicherschutz und in der
- Kapselung der Benutzerprozesse durch Verwaltung der Speicher-Abbildung nur durch Kern-Prozesse.

Durch diese Mechanismen wird erreicht:

- Vollständige Isolation der Speicherbelegung von User-Prozessen untereinander und gegen den Kern.
- Verwaltung der Speicher-Belegung nur durch den Kern.
- Abwicklung der Kern-Programme im priviligierten Supervisor-Mode.

Diese Mechanismen können nicht umgangen werden.

Mechanismen zur Kontrolle der Schnittstelle:

Die Mechanismen bestehen aus der

- Prüfung der durch das aufrufende Programm übergebenen Parameter im Kern durch den Systemaufruf,
- Prüfung der Benutzeridentität und -privilegien in Systemaufrufen, die besondere Rechte erfordern und
- Prüfung der Zulässigkeit der Speicheradressen.

Durch diese Mechanismen wird kontrolliert, daß die Schnittstelle zum Kern nur in der dokumentierten Weise genutzt werden kann.

Die genannten Prüfungen in den Systemaufrufen erfolgen in Abhängigkeit vom jeweiligen Aufruf und der Art der jeweiligen Parameter.

3.4.2 Mechanismen zur Abgrenzung der Subsysteme

Die Abgrenzung der Subsysteme ist nur gewährleistet, wenn die Integrität des Gesamtsystems bezogen auf SGID-Programme, Benutzer-Eintragungen, Vergabe von Privilegien, Eigentum und Rechte an allen Einheiten des Subsystems durch die Administratoren gesichert wird.

3.4.3 Mechanismen zur Abgrenzung der übrigen SUID / SGID-Programme

Die Abgrenzung der SUID / SGID-Programme, die nicht zu einem Subsystem aber zum sicherheitsrelevanten Teil des Systems gehören, geschieht zum einen über die Mechanismen zur Rechteverwaltung:

- Für die Programmdateien der SUID / SGID-Programme ist nur das Ausführungsrecht gesetzt. Damit sind weder lesende Zugriffe, z.B. über den Debugger, noch unbefugte Modifikationen möglich.
- Der Aufrufer muß das *execsuid* Privileg besitzen.

- Falls eine Programmdatei überschrieben wurde, muß nach dem Überschreiben der Programmdatei das s-Bit wieder explizit vom *Super User* vergeben werden.

Zum anderen sind die mit dem System gelieferten SUID / SGID-Programme vom Hersteller besonders überprüft worden, um Sicherheitslücken in der Implementation auszuschließen.

3.4.4 Mechanismen zur Abgrenzung der unbenutzten Treiber

Durch die in den Treibern eingebauten Protokolle können diese während der Boot-Phase zweifelsfrei feststellen, ob die ihnen zugeordnete Hardware vorhanden ist oder nicht. Sind die zugehörigen Hardwarekomponenten nicht vorhanden, werden die Treiber so initialisiert, daß Aufrufe über die Einträge in */dev* abgewiesen werden.
Da die Treiber nach Einstecken der notwendigen Hardware beim Booten aktiviert werden, muß der Zugriff von Unbefugten zur Maschine durch organisatorische Maßnahmen unterbunden werden.

3.5 Qualität des Herstellungsvorgangs

Gemäß den Forderungen der IT-Kriterien (Q2) müssen

- die auf der niedrigsten Hierarchiestufe der Spezifikation definierten Funktionseinheiten identifizierbar und ihre Schnittstellen untereinander dokumentiert sein,
- eine Bibliothek mit Testprogrammen und zugehöriger Testdokumentation zur Verfügung stehen und
- mittels der Durchführung von Tests Unklarheiten und vermutete Schwachstellen, die sich bei der Prüfung der Spezifikation ergaben, aufgeklärt werden.

Als Funktionseinheiten können die Systemaufrufe betrachtet werden. Diese sind in der Spezifikation durch die Beschreibung der Aufrufschnittstelle im Manual, die Einsprungtabellen in den Kerncode und den zugehörigen Funktionscode identifizierbar.
Die Schnittstellen (Abhängigkeiten, Wechselwirkungen) der Systemaufrufe untereinander sind in den Manualen dokumentiert.

Bei den Tests ist zu unterscheiden zwischen denen, die das Basissystem, und solchen, die das *"Security Package*" betreffen.
Für das *Security Package* stand eine Bibliothek von Testprogrammen mit detaillierter Testdokumentation (Testplan, Zweck, Vorgehensweise, Ergebnisse) zur Verfügung.

Sicherheitsfunktionen, die im Basissystem realisiert sind, wurden - soweit die zugehörigen Funktionsaufrufe Bestandteil des X/OPEN-Standard sind - im Rahmen der *"X/OPEN System Verification Suite*" überprüft. Die Tests sind in Manual-Format beschrieben, getrennt nach ANSI, POSIX, X/OPEN u.a. .
Von den Evaluatoren wurden darüber hinaus Tests durchgeführt, die nicht in der Testbibliothek enthalten sind.

Beim Test der Systemaufrufe mit unzulässigen Parametern ergaben sich z.T. Effekte, die nicht konform mit der Beschreibung der Spezifikation (in diesem Fall den *System Call Manuals*) sind. Sicherheitslücken, d.h. Verstöße gegen die Sicherheitsanforderungen, wurden dabei nicht festgestellt.
Bei Tests, die zur Aufklärung von Unklarheiten oder vermuteten Schwachstellen durchgeführt wurden, wurden keine Wege gefunden, die Sicherheitsfunktionen außer Kraft zu setzen.

Einschränkend ist zu bemerken, daß Tests nur soweit durchgeführt werden konnten, wie es in Bezug auf Aufwand und System-Konfiguration im Rahmen der Q2-Evaluierung erforderlich und möglich war. Fehler, die aufgrund starker Belastung, extremer Randbedingungen oder im Zusammenwirken komplexer Software-Pakete auftreten können, wurden deshalb weder gesucht noch (zufällig) gefunden.

3.6 Betriebsqualität

Die Funktionalitäten, die die Einhaltung der Sicherheitsanforderungen im laufenden Betrieb gewährleisten, erfüllen die Qualitätstufe Q2.

Konfigurierbarkeit:

Das evaluierte System sieht eine feste Software-Konfiguration vor. Es ist nicht möglich oder vorgesehen, etwa nach Baukastenprinzip zusätzliche Software-Teile zum Kern dazuzubinden. Der Kern */vmunix* enthält bereits alle Treiber für die von SNI für das MX-300 SINIX-S V5.22 System angebotenen zusätzlichen Geräte (*SCSI-Harddisk, Ethernet Controller* usw.). Bei Hinzufügen von Geräten, d.h. Änderung der Hardware-Konfiguration, werden die entsprechenden Treiber beim Hochfahren aktiviert. Der Kern "erkennt" sozusagen die gegebene Hardware-Konfiguration und konfiguriert sich dynamisch durch die Aktivierung dieser Treiber. Um ein solches Gerät über die Software ansprechen zu können, muß die zugehörige Gerätedatei vorhanden sein oder angelegt werden. Nur der *Super User* kann eine solche Datei einrichten.
System-Konfigurationen, die durch das Hinzufügen von Hardware, die nicht zum evaluierten System gehört, erweitert werden, gelten als **nicht evaluiert**. Die "erlaubten" Konfigurationsmöglichkeiten beschränken sich auf den Anschluß zusätzlicher Terminals oder eines Druckers an die serielle Schnittstelle.

Generierung:

Eine System-Generierung findet bei SINIX-S V5.22 nicht statt. Der Kern und die übrigen zu evaluierenden Systemteile werden als ausführbare Objekte ausgeliefert, die lediglich installiert werden. Im *Security Features Administrator's Guide* wird darauf verwiesen, daß während der Installation kein anderer Benutzer Zugang zum System haben darf (Ein-Benutzer-Modus). Die Vorgänge während der Installation werden in einer nur dem *Super User* zugänglichen Datei protokolliert.

Übertragungsfehler:

Beim Einspielen von Software vom Streamer-Tape erfolgt eine Überprüfung mittels eines CRC-Verfahrens. Stimmen die berechneten Prüfsummen und die Prüfsummen auf dem Band nicht überein, bricht das System den Lesevorgang ab und meldet einen Lesefehler.

Schutz vor Manipulation der Software auf dem Übertragungsweg:

Bei Installation des Systems SINIX-S V5.22 werden zu jeder eingespielten Datei die Dateigröße, die Zeit des letzten Zugriffs und der letzten Änderung, sowie eine Prüfsumme ausgegeben und in die Datei */C2install.out* protokolliert.
Der Anwender kann sich durch den Vergleich der protokollierten Prüfsummen mit den unter dem Gliederungspunkt 1.1 dieses Berichtes aufgelisteten Prüfsummen davon überzeugen, daß er das evaluierte Produkt installiert hat.

Hardware- / Software-Wartung:

In SINIX-S V5.22 gibt es keinen besonders privilegierten Wartungsmodus oder dergleichen. Die Wartung des laufenden Systems erfolgt unter einer normal-privilegierten Kennung *tele*. Dabei ist zu beachten, daß auch für diese Kennung das ausgelieferte Paßwort geändert werden muß, es also kein Standard-Wartungs-Paßwort gibt.
Spezielle Aktionen erfordern die *Super User*-Berechtigung oder können nur bei heruntergefahrenem (Hardware-Wartung) bzw. im Ein-Benutzer-Modus gefahrenem System durchgeführt werden. In jedem Fall sind keine Sicherheitsfunktionen außer Kraft gesetzt.

Sicherer Wiederanlauf nach Fehler / Wartung:

Wird das System nach der Wartung oder einem Absturz wieder hochgefahren, so befindet es sich in der Regel nach dem Booten im Mehr-Benutzer-Modus, in dem die in den Sicherheitsanforderungen dargelegten Sicherheitsfunktionen greifen (d.h. Benutzer müssen sich identifizieren und authentisieren, das Beweissicherungs-System läuft, falls dies vom *Audit Administrator* so eingestellt wurde usw.). Scheitert der Boot-Vorgang, z.B. weil das Dateisystem korrumpiert ist (infolge des Absturzes) und ein manueller *file system check* (*/etc/fsck*) notwendig ist, so wird das System in den Ein-Benutzer-Modus gebracht. Falls so eingestellt, muß sich der *Super User* mit dem *Super User* Paßwort authentisieren. Sind beim Absturz irreparable Schäden entstanden, so muß der *Super User* das System neu installieren.

Beim Booten von einer Diskette gelangt der Benutzer automatisch in den *Super User* Status. Dabei wird aber nicht der zum evaluierten System gehörige Kern aktiv. Ausgehend von dem gebooteten System ist es aber möglich, über *mount* auf

die Dateisysteme des evaluierten Systems zuzugreifen. Der Versuch, Programme des *Security Package* (z.B. *passwd*) auszuführen, wird abgewiesen. Ein direkter Zugriff auf geschützte Dateien ist allerdings möglich.
Durch organisatorische Maßnahmen ist dafür zu sorgen, daß normale Benutzer keinen Zugang zur Zentraleinheit erhalten.

Selbsttesteinrichtungen der Hardware / Firmware:

Beim Hochfahren führt das System folgende Selbsttests durch:

- Test des local DUART (*Dual Universal Asynchronous Receiver / Transmitter*)
- Test des dynamischen RAM
- Abfrage, ob die Hardware-Uhr läuft
- Test der Multibus-Fenster Einstellung
- NMI-Tests (*Non Maskable Interrupt*) NMI-Register, Bustimeout- und Parity-NMI werden getestet
- Test des Multibus-Mapper-RAM
- Test der MMU (*Memory Management Unit*)
- Test der ICU (*Interrupt Control Unit*)
- Test der FPU (*Floating Point Unit*)

Es wird also für die wesentlichen Hardware-Komponenten ein Selbsttest durchgeführt.

3.7 Qualität der Anwenderdokumentation

In der englischsprachigen, sicherheitsbezogenen Anwenderdokumentation

- "*MX-300 Security Features: User's Guide*", June 1990
- "*MX-300 Security Features: Administrator's Guide*", June 1990

wird die Handhabung der Sicherheitsfunktionen des SINIX-S V5.22 Systems für die Qualitätsstufe Q2 ausreichend umfangreich, verständlich und handhabbar beschrieben.
Die mit dem SINIX-S V5.22 System ausgelieferte Anwenderdokumentation ist dem Gliederungspunkt 1.2 (Liste der zum evaluierten System gehörenden Anwenderdokumentation) zu entnehmen.

4. Weitere Hinweise und Auflagen

Wesentliche Vorausetzung für die Gesamtsicherheit des Systems ist es, daß für alle Administrationsaufgaben nur Personen mit guten Systemkenntnissen ausgewählt werden.

Folgende Hinweise und Auflagen sind zu beachten:

- Die Zentraleinheit ist so aufzustellen, daß sie für einen normalen Benutzer physisch nicht zugänglich ist.
- Die Terminals privilegierter Benutzer sind so aufzustellen, daß sie für einen normalen Benutzer physisch nicht zugänglich sind.
- Mit organisatorischen Maßnahmen muß sichergestellt sein, daß die Hardware-Konfiguration nicht durch unbefugte Benutzer geändert werden kann und daß Änderungen durch den *Super User* nur in dem für das evaluierte System zulässigen Rahmen erfolgen. Die erlaubten Konfigurationsmöglichkeiten beschränken sich auf den Anschluß zusätzlicher Terminals oder eines Druckers an die serielle Schnittstelle.
- Während der Installation darf kein anderer Benutzer als der *Super User* Zugang zum System haben.
- Nach der Installation des Systems sollten die Dateibestände und -prüfsummen anhand der Datei /C2install.out mit dem im Gliederungspunkt 1.1 hinterlegten Stand verglichen werden, um auf diese Weise zu überprüfen, ob es sich um den evaluierten Softwarestand handelt.
- Datenträger mit Systemprogrammen (Sicherungen, Boot-Software) sollten so verwahrt werden, daß sie für einen normalen Benutzer physisch nicht zugänglich sind.
- Nach dem Lesen oder Beschreiben von Bändern oder Disketten sind diese aus dem Laufwerk zu entfernen und sicher zu verwahren, da sie sonst allgemein zugreifbar sind.
- Der *Super User* und die privilegierten Benutzer *Authentication Administrator* bzw. *Audit Administrator* müssen vertrauenswürdige Personen sein.
  Um das Gefahrenpotential, das von den Privilegien des *Super User*s ausgeht, gering zu halten, ist durch organisatorische Maßnahmen zu

gewährleisten, daß möglichst wenige, aber mehr als eine Person in die Lage versetzt werden, die Rolle des *Super Users* einzunehmen.

- Der *Super User* und die privilegierten Benutzer *Authentication Administrator* bzw. *Audit Administrator* müssen den "*Security Features Administrators Guide*" lesen und beachten.
- Der *Super User* und die privilegierten Benutzer *Authentication Administrator* bzw. *Audit Administrator* sollten für ihre Administrationsaufgaben die entsprechenden Menüsysteme benutzen.
- Der *Super User* darf nach der Installation des Systems an den sicherheitsrelevanten Dateien keine Rechteänderungen vornehmen.
- Die Zugriffsrechte an Gerätedateien mit Dateisystemen müssen so gesetzt sein, daß es normalen Benutzern nicht möglich ist, durch direkten Zugriff auf die Gerätedatei auf die darin gespeicherten Daten des Dateisystems zuzugreifen.
- Der *Super User* ist angewiesen, auf Dateisysteme nur mit den dafür vorgesehenen Kommandos und Funktionen zuzugreifen.
- Der *Super User* darf keine zusätzlichen SUID / SGID-Programme mit privilegiertem Eigentümer einspielen oder einrichten.
- Der *Super User* oder authorisierte Administratoren dürfen die sicherheitsrelevanten Teile des Systems nicht in Bezug auf die sie definierenden Eigenschaften modifizieren. Mit dem Programm *integrity* ist die Integrität der sicherheitsrelevanten Teile des Systems regelmäßig zu überprüfen.
- Der *Super User* sollte Bänder und Disketten mit Software oder sensitivem Inhalt nur im Ein-Benutzer-Modus schreiben bzw. lesen.
- Der *Super User* darf keine ihm unbekannten Programme unter seiner Kennung ausführen ("Trojanisches Pferd").
- Verzeichnisse, in denen Programmdateien gehalten werden, die auch von privilegierten Benutzern ausgeführt werden, dürfen für normale Benutzer nicht beschreibbar sein.
- Die Beweissicherung ist so einzustellen, daß sie sich beim Hochfahren automatisch einschaltet.
- Die Beweissicherung ist so einzustellen, daß sie sich im Falle einer Überlaufsituation ganz abschaltet. Darüberhinaus sollte der *Super User* oder der *Audit Administrator* auf bevorstehende Überlaufsituationen achten und

ggf. die nötigen Schritte (Sicherung der protokollierten Daten) unternehmen.

- Die Beweissicherung ist so einzustellen, daß sowenig Protokollsätze wie möglich bei einem Systemabsturz verloren gehen. Die Angabe eines kleinen Wertes kann aber die Systemperformance entscheidend beeinflussen. Der Anwender muß selbst entscheiden, wie wichtig ihm der Schutz der Protokolldaten im Falle eines Systemabsturzes ist.
- Das Authentisierungs-System ist so einzustellen, daß die von den Benutzern gewählten Paßwörter geprüft werden. Die Liste mit "schlechten" Paßwörtern ist dazu betreiberspezifisch zu erstellen.
- Das Paßwort der normal-privilegierten Wartungskennung *tele* ist nach Auslieferung des Systems zu ändern.
- Das Authentisierungs-System ist so zu konfigurieren, daß vor dem Übergang in den Ein-Benutzer-Modus eine Authentisierung stattfindet.
- Jeder Benutzer des Systems muß den "*Security Features Users Guide*" lesen, um sich mit den vorhandenen Sicherheitsmechanismen vertraut zu machen.
- Die Benutzer sollten die Rechte an ihren Objekten so vergeben, daß nur solche Benutzer Zugriff haben, für die das gewünscht wird. Dies gilt speziell für die Rechte an Verzeichnissen.
  Die Vergabe des Schreibrechtes an Verzeichnissen an andere Subjekte als den Eigentümer ist auf das notwendige Maß einzuschränken vor allem dann, wenn diese Subjekte mit dem Systemprivileg *chown* ausgestattet sind.
- Dateien mit vertrauenswürdigem Inhalt sollten mit *destroy* gelöscht werden, wenn sie nicht mehr benötigt werden.
- Die Benutzer sind anzuhalten, beim Verlassen des Arbeitsplatzes sich am System abzumelden.

Verzeichnis der Abkürzungen

DAC *Discretionary Access Control*
Benutzerbestimmbare Zugriffskontrolle

EGID Effektive Gruppen-Kurzkennung / Gruppen-Identifikation
Anhand der effektiven Gruppen-Kurzkennung wird im System die Gruppenzugriffsberechtigung eines Prozesses gegen die Gruppenzugriffsberechtigung des Dateiobjektes geprüft, z.B. beim Öffnen einer Datei.

EUID Effektive Benutzer-Kurzkennung / Benutzer-Identifikation
Anhand der effektiven Benutzer-Kurzkennung wird im System die Benutzerzugriffsberechtigung des Prozesses gegen die Eigentümerzugriffsberechtigung des Dateiobjektes geprüft, z.B. beim Öffnen einer Datei.

LUID Login Benutzer-Kurzkennung / Benutzer-Identifikation
Die Login Benutzer-Kurzkennung stellt eine vom System vergebene eindeutige Nummer dar, die innerhalb der Beweissicherung zur eindeutigen Identifizierung eines Benutzers verwendet wird. Sie wird für jeden Benutzer vom System auf den Wert der dem Benutzer beim Eintragen in das System zugewiesenen realen Benutzer-Kurzkennung (RUID) gesetzt.

PID Prozeßidentifikation
Die Prozeßidentifikation stellt eine vom System vergebene eindeutige fortlaufende Nummer dar, anhand der der Prozeß von anderen Prozessen angesprochen werden kann.

PPID Prozeßidentifikation des Vaterprozesses
Die Prozeßidentifikation des Vaterprozesses (PPID) zeigt im Sohnprozeß die PID des Vaterprozesses an. Dieser Wert ist beim *exit()* des Sohnprozesses für die Signalübermittlung an den Vaterprozeß von Bedeutung.

PWD *Permission Mode Word*
Die Zugriffsrechte an Dateien sind in der Kontrollstruktur *vnode* jeder Datei im sogenannten *Permission Mode Word* abgelegt. Anhand dieser Information erfolgt die Rechteprüfung.

RGID Reale Gruppen-Kurzkennung / Gruppen-Identifikation
Anhand der realen Gruppen-Kurzkennung wird im System die Gruppenzugehörigkeit des Erzeugers des Prozesses identifiziert.

RUID Reale Benutzer-Kurzkennung / Benutzer-Identifikation
Anhand der realen Benutzer-Kurzkennung wird im System der Erzeuger des Prozesses identifiziert.

Auszüge aus den IT-Sicherheits-kriterien

Die nachstehenden Auszüge aus den IT-Sicherheitskriterien beschreiben die Anforderungen an das zertifizierte Produkt und verdeutlichen die erreichten Bewertungsstufen.

"Funktionalitätsklasse F1

(abgeleitet aus der Funktionalität der Orange Book Klasse C1)

Identifikation und Authentisierung
Das System muß Benutzer identifizieren und authentisieren. Diese Identifikation und Authentisierung muß vor jeder anderen Interaktion des Systems mit dem Benutzer erfolgt sein. Nur nach einer erfolgreichen Identifikation und Authentisierung dürfen andere Interaktionen möglich sein. Die Authentisierungs-informationen müssen so gespeichert sein, daß nur autorisierte Benutzer Zugriff dazu besitzen.

Rechteverwaltung
Das System muß Zugriffsrechte zwischen Benutzern und / oder Benutzergruppen und Objekten, die der Rechteverwaltung unterliegen, kennen und verwalten. Dabei muß es möglich sein, Benutzern bzw. Benutzergruppen den Zugriff zu einem Objekt ganz zu verwehren. Vergabe und Entzug von Zugriffsrechten zu einem Objekt darf nur durch autorisierte Benutzer möglich sein.

Rechteprüfung
Das System muß bei jedem Zugriffsversuch von Benutzern bzw. Benutzergruppen zu den der Rechteverwaltung unterliegenden Objekten die Berechtigung überprüfen. Unberechtigte Zugriffsversuche müssen abgewiesen werden."

"Funktionalitätsklasse F2

(abgeleitet aus der Funktionalität der Orange Book Klasse C2)

Identifikation und Authentisierung
Das System muß Benutzer identifizieren und authentisieren. Diese Identifikation und Authentisierung muß vor jeder anderen Interaktion des Systems mit dem Benutzer erfolgt sein. Nur nach einer erfolgreichen Identifikation und Authentisierung dürfen andere Interaktionen möglich sein. Die Authentisierungsinformationen müssen so gespeichert sein, daß nur autorisierte Benutzer Zugriff dazu besitzen.
Bei jeder durchgeführten Interaktion muß das System die Identität des Benutzers feststellen können.

Rechteverwaltung
Das System muß Zugriffsrechte zwischen Benutzern und / oder Benutzergruppen und Objekten, die der Rechteverwaltung unterliegen, kennen und verwalten. Dabei muß es möglich sein, Benutzern bzw. Benutzergruppen den Zugriff zu einem Objekt ganz zu verwehren sowie den Zugriff auf einen nicht-modifizierenden Zugriff zu beschränken. Außerdem muß es möglich sein, für jeden Benutzer separat die Zugriffsrechte zu einem Objekt festzulegen. Vergabe und Entzug von Zugriffsrechten zu einem Objekt darf nur durch autorisierte Benutzer möglich sein.
Die Weitergabe von Zugriffsrechten muß kontrollierbar sein. Ebenso darf das Einbringen neuer Benutzer sowie das Löschen bzw. Sperren von Benutzern nur durch autorisierte Benutzer möglich sein.

Rechteprüfung
Das System muß bei jedem Zugriffsversuch von Benutzern bzw. Benutzergruppen zu den der Rechteverwaltung unterliegenden Objekten die Berechtigung überprüfen. Unberechtigte Zugriffsversuche müssen abgewiesen werden.

Beweissicherung
Das System muß eine Protokollierungskomponente enthalten, die in der Lage ist, folgende Ereignisse mit folgenden Daten zu protokollieren:

Benutzung des Identifikations- und Authentisierungsmechanismus
Daten: Datum; Uhrzeit; Benutzer-Id., die angegeben wurde; Kennung des Gerätes, an dem der Identifikations- und Authentisierungsmechanismus benutzt wurde (z. B. Terminal-Id.); Erfolg bzw. Mißerfolg des Versuches.

Zugriff zu einem der Rechteverwaltung unterliegenden Objekt
Daten: Datum; Uhrzeit; Benutzer-Id.; Name des Objektes; Art des Zugriffsversuches; Erfolg bzw. Mißerfolg des Versuches.

Anlegen bzw. Löschen eines der Rechteverwaltung unterliegenden Objektes
Daten: Datum; Uhrzeit; Benutzer-Id.; Name des Objektes; Art der Aktion.

Aktionen von Benutzern bzw. Rollen mit besonderen Rechten, die die Sicherheit des Systems betreffen
Daten: Datum; Uhrzeit; Benutzer-Id.; Art der Aktion; Name des Objektes, auf das sich die Aktion bezog (Solche Aktionen sind z. B. Einbringen oder Löschen (Sperren) von Benutzern; Einbringen bzw. Entfernen von Datenträgern; Starten bzw. Stoppen des Systems).

Der Zugriff zu den Protokollinformationen darf nur dazu autorisierten Benutzern möglich sein. Es muß möglich sein, die Protokollierung auf einen oder mehrere beliebig wählbare Benutzer zu beschränken. Es müssen Auswerte- und Verwaltungsprozeduren für die Protokolldaten vorhanden und dokumentiert sein. Der Aufbau der Protokollsätze muß vollständig beschrieben sein.

Wiederaufbereitung
Alle Speicherobjekte müssen vor einer Wiederverwendung durch andere Benutzer so aufbereitet werden, daß keine Rückschlüsse auf ihren früheren Inhalt möglich sind."

Charakterisierung der Qualitätsstufen

Zur Bewertung der Qualität werden 8 Qualitätsstufen definiert. Dabei bezeichnet Q0 die niedrigste Stufe, Q7 die höchste definierte Qualitätsstufe. Diese Stufen beinhalten unter anderem Anforderungen an den Prüfvorgang und die dabei verwendeten Verfahren, sowie an die Stärke der im Produkt implementierten Mechanismen.

a) Die hier in Frage kommenden Qualitätsstufen lassen sich hinsichtlich des Prüfvorgangs folgendermaßen charakterisieren:

"Q1 : getestet
Die Sicherheitsanforderungen sind verbal formuliert. Die Spezifikation beschreibt relativ grob die Art der Implementierung. Das System wurde mit Hilfe einfacher Tests auf Erfüllung der Sicherheitsanforderungen geprüft. Dabei wurden keine Fehler gefunden.

Q2 : methodisch getestet
Die Sicherheitsanforderungen sind verbal formuliert. Das System wurde an Hand der Spezifikation methodisch getestet und auf Erfüllung der Sicherheitsanforderungen geprüft. Dabei wurden keine Fehler gefunden.

Q3 : methodisch getestet und teilanalysiert
Die Sicherheitsanforderungen sind detailliert verbal beschrieben. Die Systemspezifikation wurde informell analysiert und auf Konsistenz mit den Sicherheitsanforderungen geprüft. Der Quellcode wurde stichprobenartig an Stellen, die als besonders kritisch angesehen wurden, analysiert. Das System wurde an Hand dieser Analysen methodisch getestet und auf

Erfüllung der Sicherheitsanforderungen geprüft. Dabei wurden keine Fehler gefunden."

b) Hinsichtlich der Stärke der verwendeten Mechanismen sind für die Qualitätstufe Q2 folgende Auszüge aus den IT-Sicherheitskriterien zu beachten:

"...Die verwendeten Mechanismen müssen in ihrer Gesamtheit ausreichend sein, die Sicherheitsanforderungen zu erfüllen. Der zur Erfüllung einer bestimmten Sicherheitsanforderung verwendete Mechanismus bzw. die verwendete Kombination von Mechanismen darf nicht schlechter als mit "mittelstark" bewertet worden sein. Es werden keine Ausnahmenfälle zugelassen."

Zur Bewertung der Mechanismen sind sechs Stufen von "ungeeignet" bis "nicht überwindbar" definiert. Die hier zu betrachtenden Stufen sind wie folgt charakterisert:

"ungeeignet
Der Mechanismus ist nicht wirksam.

schwach
Der Mechanismus ist lediglich als Abwehr unbeabsichtigter Verstöße gegen die Sicherheitsanforderungen geeignet, stellt jedoch kein ernsthaftes Hindernis gegen absichtliche Verstöße dar.

mittelstark
Der Mechanismus bietet bereits Schutz bei absichtlichen Verstößen gegen die Sicherheitsanforderungen, ist jedoch mit mittelgroßem Aufwand von Personen mit normalen Kenntnissen des Systems zu überwinden.

stark
Der Mechanismus bietet einen guten Schutz bei absichtlichen Verstößen gegen die Sicherheitsanforderungen und ist nur mit großem Aufwand bzw. unter Zuhilfenahme aufwendiger Hilfsmittel zu überwinden. Falls organisatorische Maßnahmen zur Aufrechterhaltung der Stärke des

Mechanismus notwendig sind, müssen diese recht einfach geartet und wenig fehleranfällig sein. Bei fehleranfälligen organisatorischen Maßnahmen müssen im zu evaluierenden IT-System Methoden zur Erkennung und Vermeidung solcher Fehler implementiert sein. Die organisatorischen Maßnahmen sind im Evaluationsbericht zu beschreiben. Die Bewertung der Qualität dieser Methoden und der Korrektheit ihrer Implementierung ist dabei Teil der Evaluation."

Best.-Nr. U7735-J-Z143-1
Gedruckt im Auftrag der Siemens Nixdorf Informationssysteme AG
600 SO 7912. (750)